# Clock Control Sequence Reconstruction in the Generalized Shrinking Generator

Jan Inge Trontveit


Master's Thesis
Master of Science in Information Security
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2006

Institutt for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

# Abstract

The shrinking generator is a popular representation of pseudorandom sequence generators (PRSGs) that employ so-called irregular clocking. This familiy of PRSGs is characterized by having one or more linear feedback shift registers (LFSRs), whose clocking is controlled by the output sequence of one of the subgenerators. Such a sequence is called clock-control sequence. In this thesis, an algorithm is presented that reconstructs the clock-control sequence in the generalized shrinking generator in the presence of noise. The shrinking generator is first reduced to a step1-step$E$ generator (another class of PRSGs with irregular clocking), where $E$ depends on the maximum length of runs of zeros in the output sequence of the clocking part of the generator. Then a directed depth-first like search for optimal and suboptimal paths in the edit distance matrix corresponding to the generator is performed. The permitted path weight deviation from the optimum is determined by the noise level in the statistical model of the generator. Influence of inadequate estimation of the length of the output sequence of the clocked LFSR without irregular clocking on the level of equivalent noise is discussed. The experimental results that are presented show that the total number of candidate clock-control sequences increases moderately as the necessary clock-control sequence length increases. The advantage of this attack over the other attacks reported in the litterature is that this attack is effective even if the noise level is relatively high and that the solution is guaranteed to be found.

# Sammendrag

"Shrinking" generatoren er en populær representant for pseudotilfeldige sekvens generatorer (PTSG) som anvender såkalt uregelmessig klokking. Det karakteristiske ved denne familien av PTSG er at de har en eller flere lineært tilbakekoblede skift registre(LTSR) hvis klokking blir kontrollert av sekvensen til en av subgeneratorene. Denne sekvensen kalles klokke-kontroll sekvensen. I denne rapporten presenterer vi en algoritme som rekonstruerer klokke-kontroll sekvensen til en generalisert "shrinking" generator med tilstedeværelse av støy. "Shrinking" generatoren blir først redusert til en step1-step$E$ generator (en annen klasse av PTSG med uregelmessig klokking), hvor $E$ avhenger av den maksimale lengden på etterfølgende nuller i ut-sekvevnsen til den klokkende delen av generatoren. Deretter utføres det et retningsstyrt søk for optimale og suboptimale stier i "edit-distance" matrisen som samsvarer med generatoren. Det tillatte vekt-avviket fra det optimale bestemmes av støy-nivået i den statistiske modellen av generatoren. Innvirkningen som utilstrekkelig estimering av lengden på ut-sekvensen fra den klokkede LTSR uten uregelmessig klokking har på nivået tilsvarende støy, diskuteres. Forsøksresultane som presenteres viser at totalt antall klokke-kontroll sekvens kandidater øker moderat etter hvert som den nødvendige klokke-kontroll sekvens lengden øker. Fordelen med dette angrepet i forhold til andre angrep rapportert i litteraturen, er at dette angrepet er effektivt selv om støy-nivået er relativt høyt og at løsningen garantert vil bli funnet.

# Acknowlegdements

I would like to thank my supervisor Professor Slobodan Petrović for all his ideas and help throughout the master's thesis period. Without his guidance, this thesis would not be possible.

A big thanks also to my fellow students in room A220 at Gjøvik University College, for humorous times and inspiring talks.

A special thanks also to Irene, my friends and my family for supporting me and being patient when days have been long and the stress factor huge.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The shrinking generator [1] (Figure 1) is a popular example of a pseudorandom sequence generator with irregular clocking. The generator produces output sequences with good cryptographic characteristics (long period, high linear complexity, good statistical properties, etc.) However, if a sufficiently long prefix of the output sequence of such a generator is known, it is possible to reconstruct the initial state of the $\mathrm{LFSR}_A$ by means of a generalized correlation attack. In [2] it was shown for a general type pseudorandom generator with irregular clocking that, by making use of a special statistical model, it is possible to determine a set of candidate initial states of the clocked LFSR, which could generate the intercepted output sequence. The statistical model employs the edit distance with the constraint on the maximum length of runs of deletions. Once the set of candidate initial states is known, the attack continues by determining the clock control sequence that, together with one of the candidate initial states of the LFSR, could generate the intercepted sequence. In this thesis, we describe a deterministic method of reconstruction

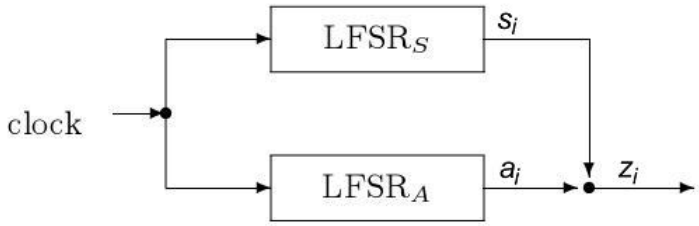


Figure 1: The shrinking generator

of clock control sequence in the generalized shrinking generator in the ciphertext only attack scenario. The shrinking generator from Figure 1 is generalized in the sense that $\mathrm{LFSR}_S$ may be replaced with a general type subgenerator. Such a generalized shrinking generator is first reduced to a step1-step$E$ generator, where $E$ depends on the maximum length of runs of zeros in the output sequence of the clocking part of the generator. Then a "depth-first"-like search through the constrained edit distance matrix associated with every candidate initial state of the $\mathrm{LFSR}_A$ is used. The paths in this matrix that correspond to candidate clock control sequences are reconstructed. Influence of noise is taken into account by relating the noise level with the permitted weight deviation from the optimum path weight used in the search process. By starting with the reconstruction of paths whose weight deviation from the optimum is 0 (the optimal paths - without noise) and then by increasing this weight deviation according to the noise level (the suboptimal paths), we make our search a directed one.

## 1.1 Preliminaries

To generate a random sequence of 0s and 1s, a coin could be tossed with a head landing up recorded as a 1 and a tail as a 0. It is assumed that the coin is unbiased, which means that the probability of a 1 on a given toss is exactly 1/2. This will depend on how well the coin is made and how the toss is performed. This method would be of little value in a system where random sequences must be generated quickly and often. It has no practical value other than to serve as an example of the idea of random number generation.

Since most true sources of random sequences (the concept of "randomness" is also questionable, but discussing this would be beyond the scope of this thesis) come from physical means, they tend to be either costly or slow in their generation. To overcome these problems, methods have been devised to construct pseudorandom[1] sequences in a deterministic manner from a shorter random sequence called a *seed*. The pseudorandom sequences appear to be generated by a truly random source to anyone not knowing the method of generation. Often the generation algorithm is known to all, but the seed is unknown except by the entity generating the sequence. A plethora of algorithms has been developed to generate pseudorandom bit sequences of various types.

A pseudorandom sequence generator (PRSG) is a deterministic[2] algorithm which, given a truly random binary sequence of length $k$, outputs a binary sequence of length $l >> k$ which "appears" to be random. The input to the PRSG is called the seed, while the output of the PRSG is called a pseudorandom bit sequence or the output sequence.

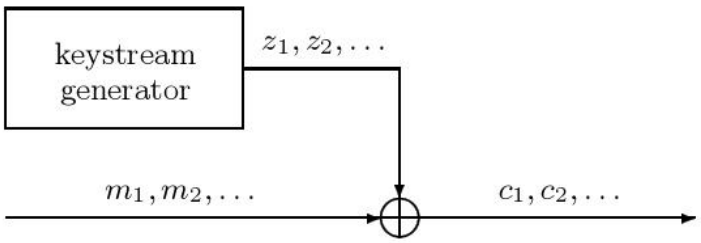


Figure 2: A PRSG used as a keystream generator in a simple stream cipher

Pseudorandom sequence generators are often used in cryptography for enciphering/deciphering (although their use is not limited to this). Such a generator is used as a keystream generator in a cipher system (Figure 2).

When used in cryptography, the output sequence of a PRSG should have at least the following properties

- long period
- good statistical properties and
- large linear complexity.

[1] From Clue: Pseudo- is used to form nouns and adjectives that refer to or describe things that are not really what they seem or claim to be

[2] *Deterministic* here means that given the same initial seed, the generator will always produce the same output sequence

An LFSR with length L and a primitive connection polynomial of degree L is often referred to as a maximum length LFSR [3, 4]. This means that such an LFSR will produce an output sequence, called an m-sequence, of maximum length, i.e. the period of the LFSR will be of length $2^L - 1$. Thus, one could say that a maximum-length LFSR is capable of generating an output sequence with *long period*. In the rest of this thesis, when referring to LFSR, we mean LFSR with period of maximum length.

Golomb's randomness postulates establish some *neccessary* conditions for a periodic pseudorandom sequence to look random. The first postulate states that in a periodic binary sequence, the number of 1s differs from the number of 0s by at most 1. The second postulate states that of the total number of runs[3] in a periodic binary sequence, approximately half of the runs have length 1, approximately one-fourth have length 2, approximately one-eigth have length 3, etc. The output sequence of an LFSR also satisfies Golomb's randomness postulates. Thus, one could say that the output sequence also has *good statistical properties*, i.e. it looks random.

The last desireable property of a PRSG is that it should also have large linear complexity (LC). LC is defined as the length of the smallest LFSR capable of generating the given output sequence. LC is a measure of unpredictability. The Berlekamp-Massey algorithm [4] may be used to calculate the LC of a binary sequence (if LC is not too large, since the Berlekamp-Massey algorithm has quadratic time complexity). The algorithm also determines the characteristics of the LFSR, namely the connection polynomial and the initial state. What makes an LFSR unsuitable for the generation of pseudorandom sequences in cipher systems, is that by using the Berlekamp-Massey algorithm we need only a relatively few consecutive bits of the LFSR's output sequence in order to easily predict the rest of the sequence.

Although LFSRs cannot be used directly as a PRSG in cipher systems, they are popular building blocks in PRSGs. The goal when designing LFSR-based PRSG is to increase the LC of the LFSR's output sequence to a level that is too high for the Berlekamp-Massey algorithm, while preserving the good statistical properties. There are various approaches to this. One of them, which we analyse in this thesis, is to use PRSG with irregular clocking. These are constructions where one or more LFSRs are used to control the clocking of one or more other LFSRs.

An example of a typical clock controlled PRSG is depicted in Figure 3. $LFSR_S$ is used to control the clocking of $LFSR_A$. If the current bit from $LFSR_S$ is 0 then $LFSR_A$ is clocked regularly, i.e. stepped once. If the current bit from $LFSR_S$ is 1 then $LFSR_A$ is clocked twice, i.e. one bit from $LFSR_A$ is deleted or omitted.

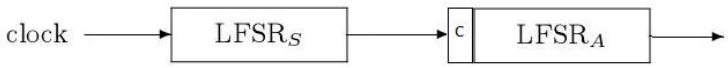


Figure 3: The step1-step2 generator

The generator from Figure 3 is easily generalized by using more than one bit of the output of the $LFSR_S$ (for example by combining bits from several positions of the $LFSR_S$) for clocking. In this thesis, we call such a generalized generator the step1-stepE gen-

[3]Let s be a binary sequence. A *run* of s is a subsequence of s consisting of consecutive 0s or consecutive 1s which is neither preceded nor succeeded by the same symbol.

erator, where $E$ is the maximum number of skipped bits from the output sequence of $LFSR_A$.

The shrinking generator is another popular example of a PRSG with irregular clocking. It is a simple construction that consists of two linear feedback shift registers (LFSRs), $LFSR_A$ and $LFSR_S$. If the output sequence from $LFSR_A$ is $\mathbf{a} = a_1, a_2, \ldots$, and the output sequence from $LFSR_S$ is $\mathbf{s} = s_1, s_2, \ldots$, then the output sequence $\mathbf{z} = z_1, z_2, \ldots$ of the shrinking generator is the sequence obtained from $\mathbf{a}$ by removing all $a_i$'s for which $s_i = 0$. Let $l_A$ and $l_S$ be the lengths of the $LFSR_A$ and $LFSR_S$ respectively. If $\gcd(l_A, l_S) = 1$ then the output sequence $\mathbf{z}$ of the shrinking generator has period $(2^{l_A} - 1)(2^{l_S - 1})$. The linear complexity $LC(\mathbf{z})$ of $\mathbf{z}$ satisfies $l_A(2^{l_S - 2}) < LC(\mathbf{z}) \leq l_A(2^{l_S - 1})$, see [4].

There are many PRSGs with high linear complexity that are still vulnerable to correlation attacks. The basic idea behind a correlation attack is to identify correlation between the output of the generator and the output of one of it's LFSRs. One way of measuring correlation between two sequences is by using the Hamming distance. However, the Hamming distance is used to compare sequences of the same length. If irregular clocking is employed in a PRSG, one should determine correlation (more precisely, generalized correlation in this case), by comparing two (or more) sequences of different lengths. Since Hamming distance cannot be used for this, using the Levenshtein Distance is an alternative. The Levenshtein or edit distance may be defined as the minimum number of elementary operations (insertions, deletions and substitutions) required to transform one sequence $X$ of length $N$ into another sequence $Y$ of length $M$. The dynamic programming approach is a classical solution for computing the edit distance matrix, where the distances between longer and longer prefixes of the sequences are successively evaluated until the final result is achieved. When applying an edit distance attack to a PRSG and depending on the generator design, some edit operations may be restricted. In this case, a so-called constrained edit distance may be necessary [5]. An effcient dynamic programming algorithm for the computation of the constrained edit distance was given in [2].

# 2 Previous work

When attacking clock-controlled generators like the shrinking generator (SG) the goal is to recover the generator's initial settings. In the case of the SG, these settings include both initial states of the LFSRs in the design. The attack can be divided into two phases. In the phase one the goal is to recover the initial state of the clocked LFSR. In the second phase the goal is to recover the clock-control sequence and consequently the initial state of the clocking subgenerator. A commom assumption by most authors is that the connection polynomials are known and that they posess a noiseless segment of the output sequence (i.e. the attack is in the known-plaintext scenario).

Several approaches to the problem of reconstructing the initial settings of the SG and the related generators can be found in the litterature. One approach, described in [6, 7] is the coding theory approach. In [6] two fast techniques for reconstructing the initial state of the clock- control shift register in the SG, one based on list decoding and the other on information set decoding, are investigated. Both start from computing the a'posteriori probabilities for the clock-control bits from a known segment of the output sequence, under the assumption that the initial state of the clock-controlled shift register is already recovered. The influence of noise is not considered, e.g. techniques are developed in the known-plaintext attack scenario.

In [7] a MAP[1] decoding technique is considered. Correlation attacks on the SG and the Alternating Step Generator based on an identified relation to the decoding problem for the deletion channel and the insertion channel, respectively, are presented. Several ways of reducing the decoding complexity are proposed and investigated, resulting in "divide-and-conquer" attacks on the two generators having considerably lower complexity than previously known attacks. The influence of noise is not considered.

[8] describes methods for fast correlation attacks, based on an interesting observation, namely that one can identify an embedded low-rate convolutional code in the code generated by the LFSR sequences. This embedded convolutional code can then be decoded with low complexity, using the Viterbi algorithm(see for example [9]). From the result of the decoding phase, the secret key can be obtained. These algorithms provide a remarkable improvement over previous methods, however they are also developed in the known-plaintext attack scenario.

In [10] a relatively new an interesting linear consistency attack, with lower complexity than previous, on a general model for clock controlled generators is presented. The attack is tested in software and confirmed as having a very low running time that follows the expected complexity $O(2^{l_s})$, where $l_S$ is the length of the clocking register. Thus the run time complexity is independent of the degree/length ($l_A$) of the clocked register. This attack is also developed in the known-plaintext attack scenario, however, the authors indicate that if they modify the algorithm it will be able to handle noise. Other approaches based on the Linear Consistency Test (LCT) [11] are presented in [12, 13].

In [14] a correlation attack using a non-linear boolean filter is described. The at-

[1] Maximum a'posteriori probability

tack uses a correlation property of Boolean functions, that gives higher correlation than previous methods. The influence of noise is not considered.

In [15] a probabilistic analysis of the shrinking generator, which shows that this generator can be vulnerable to a specific fast correlation attack is conducted. The first stage of the attack is based on a recursive computation of the posterior probabilites of individual bits of the regularly clocked $LFSR_A$ sequence when conditioned on a given segment of the output sequence. Theoretical analysis shows that these probabilities are significantly different from one half and can hence be used for reconstructing the initial state of $LFSR_A$ by iterative probabilistic decoding algorithms for fast correlation attacks on regularly clocked LFSR's. In the second stage of the attack, the initial state of $LFSR_S$ is reconstructed in a similar way, which is based on a recursive computation of the posterior probabilites of individual bits of the $LFSR_S$ sequence when conditioned on the output sequence and on the reconstructed $LFSR_A$ sequence. The influence of noise is not considered.

In [16] a probabilistic correlation attack on irregularly clocked shift registers is applied in a divide and conquer attack on the shrinking generator. Systematic computer simulations show that the joint probability is a suitable basis for the correlation attack and that, given an output sequence segment of length linear in the length of the clock-controlled shift register, the shift register initial states can be identified with high probability. This approach is also developed in the known-plaintext attack scenario.

In [17], the possibility of clock control sequence reconstruction by backtracking through the edit distance matrix is mentioned in the context of cryptanalysis of the alternating step generator. The method used in [17] considers the known plaintext attack scenario, i.e. with the noise probability equal to zero.

In [5] a new algorithm based on two different and independent ways to improve a known constrained edit distance attack on clock-controlled LFSR-based generators is proposed. The described algorithm avoids the exhaustive search over all the initial states of the involved LFSRs. The most remarkable aspect of this work is that the general ideas that have been proposed may be applied to attack any clock-controlled LFSR-based stream cipher. Again, the known-plaintext attack scenario is studied.

In the process of reconstructing the clock-control sequence, the influence of noise on the effectiveness of the procedure applied is decisive. Namely, the noise can either prevent a procedure from functioning or significantly reduce its effectiveness. Besides, the ciphertext-only attack on stream ciphers, as the most difficult one, is the most realistic attack scenario.

In [2] a generalized correlation attack determining a set of candidate initial states of the clocked register is described. A special statistical model that employs the Constrained Levenshtein or edit distance with the constraint on the maximum number of consecutive deletions is used.

In [18], a clock-control sequence reconstruction procedure capable of handling noise (ciphertext-only attack scenario) is presented. In the phase one, candidate initial states of the clocked register are reconstructed by means of the procedure given in [2]. In the phase two, the edit distance matrices creaated in the phase one are used to reconstruct the clock-control sequence. The procedure is applied on the step1-step2 generator and results from these experiments show that the average number of paths that have to be recontructed increases moderately with the level of noise. The procedure uses a depth

first like search through the edit distance matrix, when recontructing the paths. In [19] an attempt to improve this search process by the use of a k-best-paths algorithm is described.

# 3 The attack

## 3.1 Reduction to the step1-step$\mathsf{E}$ generator

The SG is a member of the familiy of clock-controlled generators. The procedure in [18] uses a statistical model (see Figure 4) of the basic structure of this family. The statistical model employs the constrained edit distance. One could readily observe that this model could be used to describe the step1-stepE generator. However, it is not that trivial to observe how the model could be used to describe the SG. In [1], where the SG was first introduced, it was stressed that the SG is not a special case of a clock-controlled generator. Intuitively, this seems reasonable when considering the SG as depicted in Figure 1. The two LFSRs are both clocked regularly by the same clock.

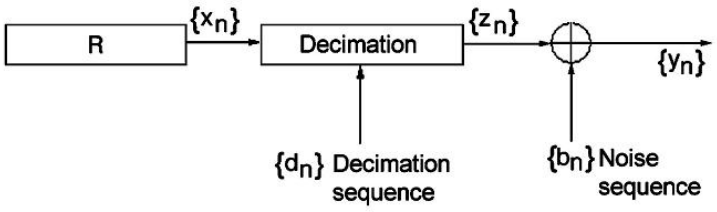


Figure 4: The statistical model of the analyzed generator

In Figure 5 we depict a scheme of a generator equivalent to the SG. It is a special case of the step1-stepE generator. The output from $\mathrm{LFSR_S}$ runs through a filter which generates the clock control sequence used to control the clocking of $\mathrm{LFSR_A}$.

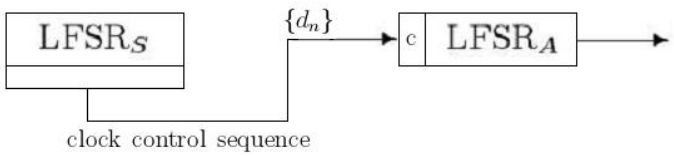


Figure 5: The step1-stepE generator

The clock control sequence $\{\mathrm{d_n}\}$ is a sequence of integers. Let $\mathrm{s}$ denote a subsequence of the output sequence S of $\mathrm{LFSR_S}$. In Table 1 one could see how $\{\mathrm{d_n}\}$ is constructed and how $\mathrm{LFSR_A}$ is clocked according to it.

The step1-stepE generator is closely related to the step1-step2 generator. Both generators delete or omit bits from the $\mathrm{LFSR_A}$ output sequence. The step1-step2 will never ommit more than $\mathsf{E} = 1$ consecutive bits. Let $\mathrm{l_S}$ denote the length of $\mathrm{LFSR_S}$. The special case of the step1-stepE generator that is equivalent to the SG will at maximum delete $\mathrm{l_S} - 1$ consecutive bits.

To illustrate the equivalence of the SG and the step1-stepE special case generator from Figure 5, let us consider the following output sequence from $\mathrm{LFSR_S}$ in the SG.

Table 1: The step1-step$E$ clock-control sequence

| $\{d_n\}$ | $s$ | step |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 01 | 2 |
| 2 | 001 | 3 |
| 3 | 0001 | 4 |
| 4 | 00001 | 5 |

$$\mathbf{s} = (0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1)$$

This would correspond to following clock-control sequence in the step1-step$E$

$$\mathbf{d} = (2, 0, 0, 1, 2, 3)$$

Let **a** be the output sequence of $LFSR_A$

$$\mathbf{a} = (0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1)$$

Now, let the SG select which bits to become part of the output sequence **z** according to **s**

$$\mathbf{z}_{SG} = (0, 0, 0, 0, 1, 1)$$

Then, let the step1-step$E$ decimate **a** according to **d**

$$\mathbf{z}_{1-E} = (0, 0, 0, 0, 1, 1)$$

The output sequences $\mathbf{z}_{1-E}$ and $\mathbf{z}_{SG}$ are equal. From Figure 1 one could readily observe that the statistical model (see Figure 4) can be used to describe the step1-step$E$. Assuming the step1-step$E$ and the SG are equal one could assume that the procedure in [18] could be applied to the cryptanalysis of the SG.

The goal for a cryptanalyst is to reconstruct the initial settings of the generator. These settings include both initial states of the internal LFSRs. In phase two of the procedure, the clock-control sequence is reconstructed. The clock-control sequence can then easily be transformed into a sequence of bits corresponding to the initial state of the clocking register. Let **d**= (2, 0, 1, 3, 0, 1) denote a reconstructed clock-control sequence. According to Table 1, this sequence is transformed into the binary sequence **b** =(0,0,1,1,0,1,0,0,0,1,1,0,1). If $l$ is the length of clocking register then the initial state of the clocking register are the first $l$ bits of **b**.

## 3.2 Determining candidate initial states

The register $R$ in the statistical model corresponds to the $LFSR_A$ from Figure 5 without irregular clocking, and the plain text sequence is modelled by the noise sequence.

Let $\{x_n\}$ be the binary sequence produced by the shift register R. Let $\{d_n\}$ be a sequence of integers, named decimation sequence, $0 \leq d_n \leq E$, where $E$ is given in advance. In the decimation process, the sequence $\{z_n\}$ is obtained in the following way:

$$z_n = x_{f(n)}, \quad f(n) = n + \sum_{i=0}^{n} d_i, \quad n = 0, 1, 2, \ldots \tag{3.1}$$

In the statistical model, it is supposed that $\{d_n\}$ is the realization of the sequence $\{D_n\}$ of independent and identically distributed (i.i.d.) random variables, with the probability $Pr(D_n = i) = \frac{1}{E+1}$, $0 \leq i \leq E$, $\forall n$.

The binary noise sequence, $\{b_n\}$, is the realization of the sequence of random i.i.d. variables $\{B_n\}$ with the probability $Pr(B_n = 1) = p < 0.5$, $\forall n$, where $p$ is the correlation parameter.

The cryptanalyst possesses $M$ consecutive bits of the sum modulo 2 $\{y_n\}$ of the decimated sequence $\{z_n\}$ and the noise sequence $\{b_n\}$. His/her task is to determine the initial state of the generator that produced the $M$ intercepted bits of the sequence $\{y_n\}$.

The correlation attack described in [2] is based on the edit distance measure with the constraint on the maximum length of runs of deletions. This distance measure is defined as follows:

Let $X$ and $Y$ be two binary sequences of lengths $N$ and $M$, respectively. Let us consider the transformation of $X$ into $Y$ using elementary edit operations – substitutions and deletions. The constrained edit distance between $X$ and $Y$ is defined as the minimum number of elementary edit operations needed to transform $X$ into $Y$, where the number of consecutive deletions is $\leq E$. Besides, the elementary edit operations are ordered in the sense that first the deletions are performed and then the substitutions. This order of elementary edit operations conforms to the real situation in the pseudorandom generator (see Figure 4).

The edit distance defined above can be determined in an iterative way, by filling the matrix of partial constrained edit distances (see, for example, [20]). For the rest of the paper, we shall use the term *edit distance matrix*, for simplicity. In the edit transformation, if $e$ represents the number of deletions and $s$ represents the number of substitutions, then the edit distance between the prefix $X_{e+s}$ of the sequence $X$ and the prefix $Y_s$ of the sequence $Y$ is given by the following expression:

$$W[e,s] = \min\{W[e-e_1, s-1] + e_1 d_e + d(x_{e+s}, y_s) \mid$$

$$\max\{0, e - \min\{N-M, (s-1)E\}\} \leq e_1 \leq \min\{e, E\}\}$$

$$s = 1, \ldots, M \qquad e = 1, \ldots, \min\{N-M, sE\}, \tag{3.2}$$

where $d_e$ represents the elementary edit distance associated with a deletion (we assume that this value is constant), $d(x,y)$ represents the elementary edit distance associated with the substitution of the symbol $x$ by the symbol $y$ and $E$ is the maximum number of consecutive deletions. From now on, we shall assume that $d(x,y) = 0$ iff $x = y$.

Any permitted sequence of elementary edit operations (i.e. the one that satisfies the given constraints) can be represented by means of a two dimensional *edit sequence* $\mathcal{S} = (\alpha, \beta)$ over the alphabet $\{0, 1, \varnothing\}$, where the 'empty' symbol $\varnothing$ is introduced in order to represent deletions, $\alpha = X$ and $Y$ is obtained by removing the empty symbols from $\beta$. The length of the sequences $\alpha$ and $\beta$ is $N$, which is the total number of deletions and substitutions in the edit transformation of $X$ into $Y$. The edit sequence is constructed according to the following rules:

1. If both $\alpha(i)$ and $\beta(i)$ are non-empty symbols, then the substitution of the symbol $\alpha(i)$ by $\beta(i)$ takes place, $1 \leq i \leq N$.

2. If $\beta(i)$ is the empty symbol, then the deletion of the symbol $\alpha(i)$ takes place, $1 \leq i \leq N$.

Having defined the basic concepts, we can now proceed with the description of the attack. The first phase of the attack consists of the following steps:

1. Since the real clock control sequence is unknown, the length $N$ of the output sequence of the LFSR $R$ without decimation has to be estimated. $N$ depends on the maximum number of consecutive deletions $E$. A proper estimation of $N$ is very important not only for the process of determining the candidate initial states of $R$, but also for the posterior phase of the clock-control sequence reconstruction. Each estimation of $N$ introduces additional noise in the statistical model. In this thesis we use the mathematical expectation of $N$ calculated on the basis of $E$. This is not an optimal estimation, since the *average* value of $E$ is much lower than $E$ in sequences satisfying the Golomb's postulates.

   Next, the threshold $T$ necessary for the classification of the initial states of $R$ should be determined, using the probability of "false alarm" $P_f$ as well as the probability of "missing the event" $P_m$, selected in advance [2]. The probability $P_f$ determines the mathematical expectation of the cardinality of the set of candidate initial states. The threshold is computed by checking $1/P_m$ initial states, selected at random. For each of them, the edit distance defined above between the output sequence generated by the actual initial state without decimation and the intercepted output sequence is calculated.

2. For every possible initial state of $R$, not used in the step 1, the constrained edit distance between its corresponding output sequence of length $N$ and the intercepted sequence of length $M$ is computed. All the initial states that produce the output sequences from $R$, whose edit distance from the intercepted output sequence is less than the threshold $T$, are included in the set of candidate initial states.

## 3.3 Clock Control Sequence Reconstruction

We call the *optimal paths* the paths through the edit distance matrix that begin at $W[N-M, M]$. Let $pl \leq M$ be the length of the clock-control sequence needed to reconstruct the initial state of the clocking subgenerator. The optimal paths pass through the cells $W[e_{p_1}, pl], \ldots, W[e_{p_n}, pl]$ in the column $pl$ of the matrix $W$, where $n$ depends on the particular sequences.

To determine the points in the column $pl$, through which the optimal paths that start at $W[N-M, M]$ pass, every cell $W[e, s]$ has, besides the value $c$ of the edit distance, four associated vectors:

1. The vector of 'primary' pointers $vp$ to the cells $W[vp[1], s-1], \ldots, W[vp[k], s-1]$ from which it is possible to arrive to the cell $W[e, s]$ with the minimum weight increment, $k \leq E+2$.

2. The vector of 'updated' pointers $vu$ to the cells $W[vu[1], pl], \ldots, W[vu[l], pl]$, through which it is possible to arrive to the cell $W[e, s]$ with the minimum weight increment (see for example [21]), $l \leq \min\{N-M+1, E(1+pl)\}$.

3. The vector of pointers $ve$ to the cells $W[ve[1], s-1], \ldots, W[ve[j], s-1]$ from which it is possible to arrive to the cell $W[e, s]$ regardless of the weight increment, $j \leq E+2$.

4. The vector of values $vj$ of the edit distances corresponding to the elements of the vector $ve$. The cardinality of this vector is also $j$.

The actual values of $k$, $l$, and $j$ depend on the particular sequences. The matrix $W$ is filled by means of the algorithm, which implements the equation (3.2) together with the updating of the four vectors mentioned above. The complete algorithm is given below.

**Algorithm 1**
**Input:**

- The sequences $X$ and $Y$ of lengths $N$ and $M$, respectively.
- The necessary length $pl$ of the clock control sequence.
- The maximum length $E$ of runs of deletions.
- The elementary distance $d_e$ associated with the deletion of a symbol.
- The elementary edit distance $d[x, y]$ associated with the substitution of the symbol $x$ by the symbol $y$, $\forall(x, y)$.

**Output:**

- The matrix $W$ of edit distances with the vectors $vp$, $vu$, $vj$, and $ve$ associated with every cell.

**comment** Initialization

$W[e, s].c \longleftarrow \infty$ , $e = 0, \ldots, N - M, s = 0, \ldots, M$ ; the vectors $vp$, $vu$, $vj$, and $ve$ associated with every cell $W[e, s]$ are empty.
$W[0, 0].c \longleftarrow 0$ ;
**comment** The row $0$ of the matrix $W$:
**for** $s \longleftarrow 1$ **until** $M$ **do**
  **begin**
    $W[0, s].k \longleftarrow 1$ ; $W[0, s].vp[1] \longleftarrow 0$ ;
    $W[0, s].c \longleftarrow W[0, s - 1].c + d[X[s], Y[s]]$ ;
  **end** ;

**comment** Main loop

**for** $s \longleftarrow 1$ **until** $M$ **do**
  **begin**
    **for** $e \longleftarrow 1$ **until** $\min\{N - M, s * E\}$ **do**
      **begin**

Let $q$ be the minimum value of the expression

$$W[e - e_1, s - 1].c + e_1 * de + d[X[e + s], Y[s]], \quad (1)$$
$$e_1 = \max\{0, e - \min\{N - M, (s - 1) * E\}\}, \ldots, \min\{e, E\}.$$

Let $n_q$ be the number of values of $e_1$ for which the expression (1) takes the value $q$. Then

$W[e,s].c \longleftarrow q$ ; $W[e,s].k \longleftarrow n_q$ ;

$W[e,s].vp$ is filled with $n_q$ values of the expression $e-e1$ corresponding to the values $e_1$ for which the expression (1) takes the value $q$.
$W[e,s].vj$ is filled with all the values of the expression (1).
$W[e,s].ve$ is filled with the values $e-e_1$ corresponding to the values of $W[e,s].vj$.

**end** ;

**comment** Determining updated pointers $vu$.

**if** $s = pl+1$ **then**

$W[e,s].vu \longleftarrow W[e,s].vp$,
$e = 0,\ldots,\min\{N-M, s*E\}$ ;

**else if** $s > pl+1$

For every element of $W[e,s].vp$, $e = 0,\ldots,\min\{N-M, s*E\}$, the elements of $W[W[e,s].vp[i], s-1].vu$, $i = 1,\ldots,W[e,s].k$ are placed into $W[e,s].vu$, deleting the repeated ones.

**end**.

Having constructed the edit distance matrix, the next step is reconstructing the candidate clock control sequences. From now on, by *paths* we mean fragments of paths that start in the column $pl$ of the matrix $W$. There are three sets of paths to be reconstructed. The first one consists of optimal paths that start at the points $e_{p_i} = W[N-M, M].vu[i], i = 1,\ldots,W[e,s].l$. The second one consists of suboptimal paths, whose weight-difference from the optimal ones is $\leq \mathcal{D}$, a threshold given in advance that depends on the noise level, that start at $e_{p_i} = W[N-M, M].vu[i], i = 1,\ldots,W[e,s].l$. The third set consists of suboptimal paths, whose weight-difference from the optimal ones is $\leq \mathcal{D}$, that start at other points in the column $pl$.

The elements of the vector $W[N-M, M].vu$ at the end of the execution of the Algorithm 1 represent the initial points of the search for the elements of the first and second set mentioned above. As for the third set, if $\mid W[e_{p_i}, pl].c - W[e, pl].c \mid \leq \mathcal{D}$, $e = 0,\ldots,\min\{N-M, sE\}$, $e \neq e_{p_i}$, for at least one $i$, then the point $W[e, pl]$ is an initial point of the search for the paths of the third set.

In order to determine the optimal and suboptimal paths that start at every initial point $\mathcal{E}$ of any set, a special depth-first like search algorithm has been devised. In this algorithm, every branching point is processed by enumerating systematically all the paths that start in it. In this search, a special kind of stack is used. A reconstructed path is rejected if at some point its weight becomes greater than the optimal weight plus $\mathcal{D}$. The complete algorithm is given below.

**Algorithm 2:**
**Input:**

- The matrix $W$ of edit distances, obtained by means of the Algorithm 1.
- The values of $pl$, $\mathcal{E}$ and $\mathcal{D}$.

**Output:**

- All the paths that start at the point $W[\mathcal{E}, pl]$ that belong to the corresponding set(s) (see text).

**comment** The $stack$ consists of the elements of the matrix $W$ together with their coordinates; the cardinality of the $stack$ is $nsp$; the current length of the reconstructed path of the edit sequence is $t$.

**comment** Initialization

$t \longleftarrow 0$ ; $e \longleftarrow \mathcal{E}$ ; $s \longleftarrow pl$ ; $nsp \longleftarrow 0$ ;

**comment** Main loop

**repeat**
$badpath \longleftarrow$ **false** ; **comment** This is the path overweight indicator
**while** $((e > 0)$ **or** $(s > 0))$ **and not** $badpath$ **do**
**begin**
**if** $(W[e, s]$ is a branching point **then**

**comment** Put $e$, $s$ and $W[e, s]$ on the $stack$

$nsp \longleftarrow nsp + 1$ ;

$stack[nsp] \longleftarrow (W[e, s], e, s)$ ;

**comment** Process a branching point

$badpath \longleftarrow$ **false** ;

**repeat**

Consider the possibility of branching from the current branching point to one of the possible successors, i.e. the point $j$. If this possibility is chosen, and after that only the branchings to the points that lead to the optimal subpaths are followed, then the total weight of the chosen subpath is

$aw \longleftarrow stack[nsp].W.vj[stack[nsp].W.j]$ ;

and the total weight of the corresponding path is

$tw \longleftarrow weight(\alpha, \beta, t) + aw$ ;

where $weight$ is the function that returns the weight of the path before the branching and $(\alpha, \beta)$ is the prefix of the edit sequence of length $t$.

**comment** $eprev$ is the value of $e$ that corresponds to the previous path element

**if** $tw <= W[\mathcal{E}, pl].c + \mathcal{D}$ **then**
$eprev \longleftarrow stack[nsp].W.ve[stack[nsp].W.j]$ ;
$stack[nsp].W.j \longleftarrow stack[nsp].W.j - 1$ ;
**if** $stack[nsp].W.j = 0$ **then** $nsp \longleftarrow nsp - 1$ ;
**until** ($eprev$ was initialized from the $stack$) **or**
(all the successors were examined) ;

**if** $eprev$ was not initialized from the $stack$ **then**
$badpath \longleftarrow$ **true** ;

```
        comment Process a non-branching point

        if (eprev was not initialized from the stack) and
           (not badpath) then
         begin
           aw ⟵ W[e, s].vj[W[e, s].j] ; tw ⟵ weight(α, β, t) + aw ;
           if tw <= W[E, pl].c + D then
             eprev ⟵ W[e, s].ve[W[e, s].j] ;
           else
             badpath ⟵ true ;
         end ;

        comment Reconstruct the current path element

        if not badpath then
         begin comment Substitution
                 t ⟵ t + 1 ; α[t] ⟵ X[s + e] ; β[t] ⟵ Y[s] ;
           for ii ⟵ 1 until e − eprev do
             begin comment The run of deletions
               t ⟵ t + 1 ; α[t] ⟵ X[e + s − ii] ; β[t] ⟵ ∅ ;
             end ;
           e ⟵ eprev ; s ⟵ s − 1 ;
         end ;
      end ;
    if not badpath then
      Store the obtained clock control sequence ;

   comment Back to the current branching point

    if nsp > 0 then
      begin
        t ⟵ t − stack[nsp].e − stack[nsp].s ;
        (e, s) ⟵ stack[nsp].(e, s) ;
      end
until nsp = 0 .
```

# 4 Experimental work

From the cryptanalytic point of view, the number of paths necessary to find the clock control sequence should be as small as possible. This number depends on $\mathcal{D}$. Given a certain level of noise in the statistical model, the behaviour of the maximum value of $\mathcal{D}$, denoted by $\mathcal{D}_{\max}$, has been analysed experimentally. The experiment has been carried out in the following way: 50 initial states of SG are chosen at random. For each of them, the output sequence corrupted by the noise sequence generated at random is produced. The noise level $p$ is the control variable of the experiment. The set of candidates for the initial state of $\mathrm{LFSR_A}$ is determined. Once the candidates have been obtained, for a fixed value of $\mathcal{D}$, the optimal and suboptimal paths are determined from the edit distance matrix corresponding to each of them. This process is repeated starting from $\mathcal{D} = 0$ and incrementing the value of $\mathcal{D}$ until the clock control sequence generated by $\mathrm{LFSR_S}$ is found. The maximum value $\mathcal{D}_{\max}$ obtained in this process is stored. At the end of the experiment, the mean value $\overline{\mathcal{D}_{\max}}$ over the values of $\mathcal{D}_{\max}$ obtained in every case is calculated. The experiment was performed on an ordinary laptop PC. The implementation of the procedure is written in C++ (see Appendix A for the most important portions of it).

## 4.1 Results

The dependence of $\overline{\mathcal{D}_{\max}}$ on $p$ for different values of $pl$ is depicted in Figure 6.

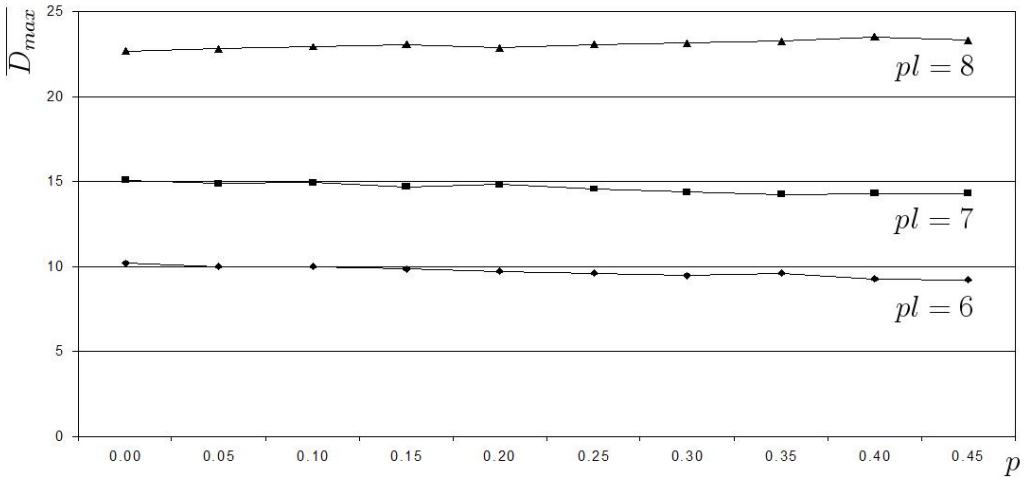


Figure 6: Dependence of $\overline{\mathcal{D}_{\max}}$ on $p$

From Figure 6 it can be concluded that:

1. $\overline{\mathcal{D}_{\max}}$ depends approximately linearly on $pl$.
2. $\overline{\mathcal{D}_{\max}}$ does not depend on $p$ at all.

Table 2: Average number of reconstructed paths for different levels of noise

| pl | $p_{0.00}$ | $p_{0.05}$ | $p_{0.10}$ | $p_{0.15}$ | $p_{0.20}$ | $p_{0.25}$ | $p_{0.30}$ | $p_{0.35}$ | $p_{0.40}$ | $p_{0.45}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 677 | 665 | 661 | 532 | 611 | 535 | 562 | 476 | 414 | 600 |
| 7 | 7652 | 7592 | 7368 | 6686 | 6536 | 6243 | 5219 | 5923 | 5948 | 4974 |
| 8 | 24549 | 24128 | 23592 | 22781 | 20564 | 19563 | 20821 | 15312 | 17703 | 18566 |

From Table 2 one could see how the average number of reconstructed paths depend on the length of necessary clock control sequence, pl.

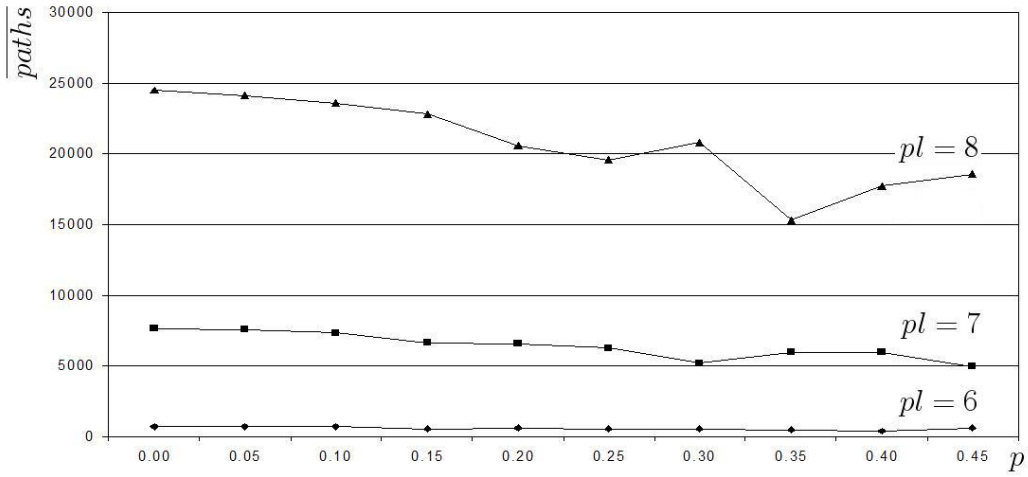


Figure 7: Average number of reconstructed paths for different values of pl and p

The experimental results presented in Figure 6 and Figure 7 are discussed in the following chapter.

# 5 Discussion

The experiment tells us that the procedure found the solution, i.e. the correct clock-control sequence, in all the tests that were performed. This means that we succeeded in applying the procedure to the SG. It tells us that it is possible to use this procedure in the cryptanlysis of the SG. We reconstructed the generator's initial settings by analyzing a prefix of its output sequence. We even succeded when we applied noise to the output sequence, meaning the procedure can operate successfully in the ciphertext-only attack scenario.

From Figure 6 we see that $\mathcal{D}_{max}$ depends approximately linearly on $\mathtt{pl}$. This is what we expected. The dependence of $\mathcal{D}_{max}$ on $\mathtt{p}$, however, is not as expected. We would expect $\mathcal{D}_{max}$ to increase as we added more noise, instead it does not seem to be affected by the noise level at all. With zero noise applied we would expect that only the optimal paths would need to be reconstructed, i.e. we would expect $\mathcal{D}_{max}$ to be zero when $\mathtt{p}$ was zero. We also expected the average number of reconstructed paths to increase as we applied more noise. In Figure 7 we see that this is not the case.

The lengths of the registers (6,7,8) we chose are small since time and computing resources don't allow trying longer LFSRs. As we can see from Figure 7, with $\mathtt{pl} = 8$ and zero noise, $\mathcal{D}_{max}$ has a value of approximately 22. This means the procedure reconstructs a lot of paths. The experiment with $\mathtt{pl} = 8$ took several days.

The reason for the "unexpected" results has to do with the estimation of N. Experiments with a fixed optimal value of N proves this (see Figure 8). We see that an optimal estimation of N would make the procedure perform significantly better. This can also be seen in the individual cases of the experiment (see Table 4) where the actual N was close to the estimated N, the value of $\mathcal{D}_{max}$ was significantly smaller. As one could see in Table 4, with LFSR's length 6, zero noise and an estimated N of 21, the value of $\mathcal{D}_{max}$ (diff) varies from 1 to to 16 depending of the actual N (n_skipped).

We expected $\mathcal{D}_{max}$ to depend on the level of noise. The experimental results tell us that $\mathcal{D}_{max}$ does not depend on $\mathtt{p}$ at all, however, it still depends on the level of noise. Inexact estimation of N is equivalent to **introducing a certain level of noise into the statistical model**. The more the estimated N differs from the "actual" N, the more "noise" is introduced. Thus, the results are as expected, however, in most of the cases there was no point in increasing $\mathtt{p}$ as the the level of noise was already at a high level due to the inexact estimation of N.

An adequate estimation of N is important and becomes more important as E increases. N is estimated as a function of E and M. E is the maximum number of consecutive deletions or the maximum length of runs of 0s in the clock control sequence. In the case of the SG, E depends on the length of the clocking register, $\mathtt{E} = \mathtt{l}_S - 1$. We know from Golomb's postulates[4] that the probability of finding a run of 0s of length E in the clock control sequence is very small. Perhaps a better solution would be to estimate N based on the average value of E which is much lower than E. However, in order to be sure we shall find the solution, we must estimate N based on a "worst case scenario". If N is

too small we shall never find the solution. If N is too large the procedure will have to reconstruct many more paths than necessary, as was the case in this experiment.

From Table 4 and the rest of the experimental results (see Appendix B) one could see that the value of $\mathcal{D}_{max}$ is almost exactly the same as the difference between the estimated N and the "actual" N (n_skipped).

Table 3: Average number of reconstructed paths for different levels of noise with optimal N

| pl | $p_{0.00}$ | $p_{0.05}$ | $p_{0.10}$ | $p_{0.15}$ | $p_{0.20}$ | $p_{0.25}$ | $p_{0.30}$ | $p_{0.35}$ | $p_{0.40}$ | $p_{0.45}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 35 | 87 | 169 | 180 | 452 | 411 | 505 | 429 | 578 | 700 |
| 7 | 224 | 527 | 924 | 3130 | 3687 | 3811 | 8678 | 5858 | 6265 | 9912 |
| 8 | 281 | 629 | 3555 | 7436 | 8334 | 5015 | 12536 | 13164 | 22623 | 30015 |

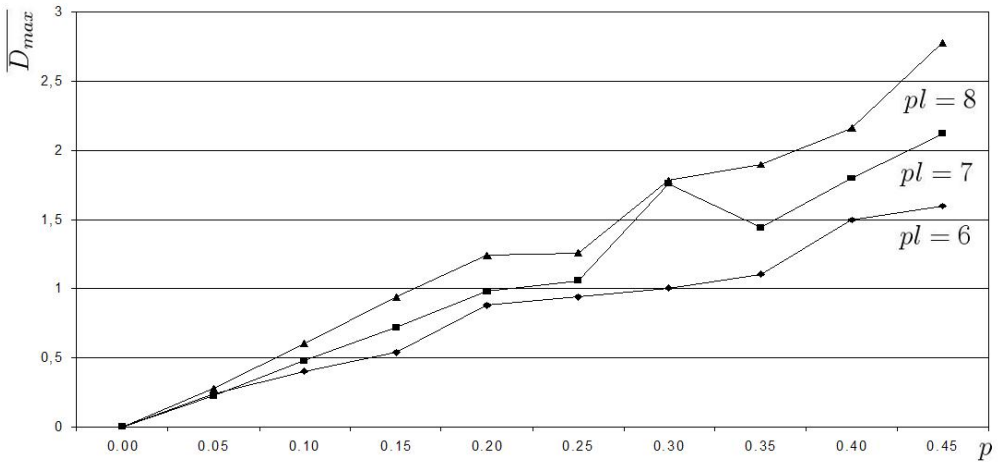


Figure 8: Dependence of $\overline{\mathcal{D}_{max}}$ on p with optimal N

Table 4: Raw data from the experiment

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.000000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 12 | 142 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 11 | 423 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 12 | 58 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 6 | 1327 | 5 | 5 | 15 | 21 |
| 6 | 0.000000 | 16 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.000000 | 3 | 1985 | 5 | 4 | 19 | 21 |
| 6 | 0.000000 | 10 | 462 | 5 | 2 | 12 | 21 |
| 6 | 0.000000 | 3 | 3685 | 5 | 5 | 18 | 21 |
| 6 | 0.000000 | 12 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.000000 | 12 | 80 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 6 | 1483 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 12 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 10 | 512 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 11 | 116 | 5 | 3 | 11 | 21 |
| 6 | 0.000000 | 1 | 250 | 5 | 4 | 20 | 21 |
| 6 | 0.000000 | 11 | 373 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 13 | 56 | 5 | 2 | 9 | 21 |
| 6 | 0.000000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 13 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.000000 | 5 | 4149 | 5 | 4 | 17 | 21 |
| 6 | 0.000000 | 13 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 11 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 12 | 126 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 14 | 7 | 5 | 1 | 8 | 21 |
| 6 | 0.000000 | 12 | 165 | 5 | 1 | 10 | 21 |
| 6 | 0.000000 | 10 | 310 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 5 | 2549 | 5 | 4 | 16 | 21 |
| 6 | 0.000000 | 13 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 8 | 836 | 5 | 2 | 13 | 21 |
| 6 | 0.000000 | 9 | 685 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.000000 | 12 | 111 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 15 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 12 | 209 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 3 | 2232 | 5 | 5 | 18 | 21 |
| 6 | 0.000000 | 6 | 2530 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 14 | 14 | 5 | 1 | 8 | 21 |
| 6 | 0.000000 | 11 | 194 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 11 | 147 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 12 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 13 | 75 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 11 | 372 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 10 | 399 | 5 | 2 | 11 | 21 |

# 6 Conclusion

In this thesis, we describe a deterministic method of reconstruction of clock-control sequence in the generalized shrinking generator in the ciphertext only attack scenario. Such a generalized shrinking generator is first reduced to a step1-step$E$ generator, where $E$ depends on the maximum length of runs of zeros in the output sequence of the clocking part of the generator. Then a "depth-first"-like search through the constrained edit distance matrix associated with every candidate initial state of the $LFSR_A$ is used. The paths in this matrix that correspond to candidate clock control sequences are reconstructed. Influence of noise is taken into account by relating the noise level with the permitted weight deviation from the optimum path weight used in the search process. By starting with the reconstruction of paths whose weight deviation from the optimum is 0 (the optimal paths - without noise) and then by increasing this weight deviation according to the noise level (the suboptimal paths), we make our search a directed one. The inexact estimation of $N$ based on the the maximum value of $E$ introduces additional noise in the statistical model. The procedure always finds the correct clock-control sequence. The maximum value of weight deviation necessary for the reconstruction of the actual clock-control sequence depends on the noise level. Experimental results show that the average number of paths that have to be reconstructed in order to find the true clock-control sequence depends on the noise level.

# 7 Future work

In this thesis we use the mathematical expectation of N calculated on the basis of the maximum length of runs of 0s, E, in the output sequence from the clocking part of the SG. This is not an optimal estimation, since the average value of E is much lower than E in sequences satisfying the Golombs postulates. A more exact estimation of N will significantly improve the procedure.

In order to further confirm the significance of the experimental results obtained in the thesis, experiments should be performed on much longer LFSRs, for example 20, 30 and even more. This would consume a significant amount of time, but it would give more weight to the conclusions of the thesis.

It would also be interesting to apply the procedure described in this thesis to the cryptanalysis of the Alternating Step Generator (ASG), a very similar generator from the same family, but still reported difficult to cryptanalyse.

# Bibliography

[1] Coppersmith, D., Krawczyk, H., & Mansour, Y. January 1994. The shrinking generator. *Lecture Notes in Computer Science*, 773, 22–.

[2] Golić, J. D. & Mihaljević, M. J. January 1991. A generalized correlation attack on a class of stream ciphers based on the levenshtein distance. *Journal of Cryptology*, 3(3), 201–212.

[3] Gollmann, D. & Chambers, W. G. May 1989. Clock-controlled shift registers: A review. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 7(4), 525–533.

[4] Menezes, A. J., Oorschot, P. C. v., & Vanstone, S. A. 1997. *Handbook of Applied Cryptography*. CRC Press.

[5] Caballero-Gil, P. & Fúster-Sabater, A. October 2005. Improvement of the edit distance attack to clock-controlled lfsr-based stream ciphers. *Lecture Notes in Computer Science*, 3643, 355–364.

[6] Chambers, W. G. & Golić, J. September 2002. Fast reconstruction of clock-control sequence. *Electronic Letters*, 38(20), 1174–1175.

[7] Johansson, T. January 1998. Reduced complexity correlation attacks on two clock-controlled generators. *Lecture Notes in Computer Science*, 1514, 342–.

[8] Johansson, T. & Jönsson, F. January 1999. Improved fast correlation attacks on stream ciphers via convolutional codes. *Lecture Notes in Computer Science*, 1592, 347–.

[9] Johannesson, R. & Zigangirov, K. S. 1999. *Fundamentals Of Convolutional Coding*. IEEE Press.

[10] Molland, H. July 2004. Improved linear consistency attack on irregular clocked keystream generators. *Lecture Notes in Computer Science*, 3017, 109–126.

[11] Zeng, K., Yang, C., & Rao, T. January 1990. On the linear consistency test (lct) in cryptanalysis with applications. *Lecture Notes in Computer Science*, 435, 164–.

[12] Zenner, E., Krause, M., & Lucks, S. January 2001. Improved cryptanalysis of the self-shrinking generator. *Lecture Notes in Computer Science*, 2119, 21–.

[13] Zenner, E. January 2003. On the efficiency of the clock control guessing attack. *Lecture Notes in Computer Science*, 2587, 200–.

[14] Molland, H. & Helleseth, T. November 2004. An improved correlation attack against irregular clocked and filtered keystream generators. *Lecture Notes in Computer Science*, 3152, 373–389.

[15] Golić, J. D. January 2001. Correlation analysis of the shrinking generator. *Lecture Notes in Computer Science*, 2139, 440–.

[16] Simpson, L., Golic, J. D., & Dawson, E. January 1998. A probabilistic correlation attack on the shrinking generator. *Lecture Notes in Computer Science*, 1438, 147–.

[17] Golić, J. D. & Menicocci, R. January 1997. Edit distance correlation attack on the alternating step generator. *Lecture Notes in Computer Science*, 1294, 499–.

[18] Petrović, S. & Fúster-Sabater, A. October 2004. Clock control sequence reconstruction in the ciphertext only attack scenario. *Lecture Notes in Computer Science*, 3269, 427–439.

[19] Herland, T. The use of k-best-paths algorithms in clock-control sequence reconstruction. Master's thesis, Høgskolen i Gjøvik, 2006.

[20] Oommen, B. 1986. Constrained string editing. *Inform. Sci.*, 40(9), 267–284.

[21] Hirschberg, D. S. 1997. Serial computations of Levenshtein distances. In *Pattern matching algorithms*, Apostolico, A. & Galil, Z., eds, 123–141. Oxford University Press.

# A Implementation in C++

```
#include "stdafx.h"

bool backtrack( int e, int s, int *n_path, int *x, int *y, int
    diff_min, int diff, mat_cell **W,
          int n_vp_max, int n_vu_max, int *solution, int E ){
  // This function reconstructs all the optimal and suboptimal
  // paths that correspond to the edit sequence of the
  // prefix X[e+s] to the prefix Y[s].
  // It only prints the paths, whose weight is between
  // optimal_weight+diff_min and optimal_weight+diff
  int i, len_prev, e_prev, b_sp, e1, s1, weight_increment,
      total_weight, counter, num_equal_bits ;
  int j, num_2 ;
  int *edit_x, *edit_y, *clock_seq ;
  branch_rec *branch_stack ;
  bool from_branch, bad_path, remove ;
  // Initialization
  branch_stack = (branch_rec *) calloc(s+1,sizeof(branch_rec)) ;
  for(i=0;i<s+1;i++){
    branch_stack[i].cell.vp = (int *)calloc(n_vp_max,sizeof(int)
        ) ;
    branch_stack[i].cell.vu = (int *)calloc(n_vu_max,sizeof(int)
        ) ;
    branch_stack[i].cell.ve = (int *)calloc(n_vp_max,sizeof(int)
        ) ;
    branch_stack[i].cell.vj = (int *)calloc(n_vp_max,sizeof(int)
        ) ;
  }
  edit_x = (int *) calloc(e+s+1,sizeof(int)) ;
  edit_y = (int *) calloc(e+s+1,sizeof(int)) ;
  clock_seq = (int *) calloc(e+s+1,sizeof(int)) ;
  len_prev = 0 ; e1 = e ; s1 = s ; b_sp = 0 ;
  branch_stack[0].e = INT_MAX ;
  branch_stack[0].s = INT_MAX ;
  // Path reconstruction
  do{
    bad_path = false ;
    while( ( (e1>0) || (s1>0) ) && (!bad_path)){
      // Check for a branching at the actual point (e,s)
      from_branch = false ;
      if( (W[e1][s1].n_vej>1) && ( (e1!=branch_stack[b_sp].e) ||
          (s1!=branch_stack[b_sp].s) ) ){
        // There is a branching, so update the branching stack
        b_sp++ ;
        branch_stack[b_sp].cell.n_vp = W[e1][s1].n_vp ;
        branch_stack[b_sp].cell.n_vu = W[e1][s1].n_vu ;
        branch_stack[b_sp].cell.n_vej = W[e1][s1].n_vej ;
```

```
    branch_stack[b_sp].cell.c = W[e1][s1].c ;
    for(i=1;i<=W[e1][s1].n_vp;i++)
      branch_stack[b_sp].cell.vp[i] = W[e1][s1].vp[i] ;
    for(i=1;i<=W[e1][s1].n_vu;i++)
      branch_stack[b_sp].cell.vu[i] = W[e1][s1].vu[i] ;
    for(i=1;i<=W[e1][s1].n_vej;i++)
      branch_stack[b_sp].cell.ve[i] = W[e1][s1].ve[i] ;
    for(i=1;i<=W[e1][s1].n_vej;i++)
      branch_stack[b_sp].cell.vj[i] = W[e1][s1].vj[i] ;
    branch_stack[b_sp].e = e1 ;
    branch_stack[b_sp].s = s1 ;
  }
  // If there is no branching, continue with the path
     reconstruction
  if( (branch_stack[b_sp].e==e1) && ( branch_stack[b_sp].s==
     s1 ) ){
    bad_path = false ;
    remove = false ;
    do{
      weight_increment = branch_stack[b_sp].cell.vj[
         branch_stack[b_sp].cell.n_vej] ;
      total_weight = weight(edit_x,edit_y,len_prev) + abs(
         weight_increment) ;
      if( total_weight<=W[e][s].c+diff ){
        // We can continue this way: select next point in
           the path
        from_branch = true ;
        e_prev = branch_stack[b_sp].cell.ve[branch_stack[
           b_sp].cell.n_vej--] ;
        // Check if the branching point is empty
        if( branch_stack[b_sp].cell.n_vej == 0 ){
          b_sp-- ;
          remove = true ;
        }
      }
      else{
        // If it is not possible to select the next point in
           the path,
        // cancel the corresponding direction
        branch_stack[b_sp].cell.n_vej-- ;
        if( branch_stack[b_sp].cell.n_vej == 0 ){
          b_sp-- ;
          remove = true ;
        }
      }
    }while(!from_branch && !remove) ;
    // If we cannot continue with the reconstruction of any
       path, raise the flag bad_path
    if (!from_branch)
      bad_path = true ;
  }
  // If there is no branching at the point (e,s),
  // continue with the path reconstruction
```

```
      if ( (!from_branch) && (!bad_path) ){
        weight_increment = W[e1][s1].vj[W[e1][s1].n_vej] ;
        total_weight = weight(edit_x,edit_y,len_prev) + abs(
            weight_increment) ;
        if( total_weight<=W[e][s].c+diff ){
          // We can continue this way: select next point in the
              path
          e_prev = W[e1][s1].ve[W[e1][s1].n_vej] ;
        }
        else{
          // If it is not possible to select the next point in
              the path,
          // cancel the corresponding direction
          bad_path = true ;
        }
      }
      // If the path is acceptable, in spite of everything, then
      // add the corresponding part of the edit sequence
      if (!bad_path){
        // Current substitution
        if(s1>0){
          edit_x[++len_prev] = x[s1+e1] ;
          edit_y[len_prev] = y[s1] ;
        }
        // Current run of deletions (if it exists)
        for(i=1;i<=e1-e_prev;i++){
          edit_x[++len_prev] = x[e1+s1-i] ;
          edit_y[len_prev] = 2 ;
        }
        // The coordinates of the point to be processed next
        e1 = e_prev ;
        s1-- ;
      }
    }
    if (!bad_path){
      // If diff_min < 0 print all the reconstructed paths ;
      // else print only the ones that have not been already
          reconstructed
      // in the previous calls to backtrack
      if ((diff_min < 0)||(weight(edit_x,edit_y,len_prev)>W[e][s
          ].c+diff_min)){
        // Reconstruct the clock control sequence from edit_x
            and edit_y
        i = len_prev + 1 ;
        counter = 0 ;
        do {
/*          i-- ;
        counter++ ;
        if (counter<=s){
          if (edit_y[i]==2){
            i-- ;
            clock_seq[counter] = 1 ;
          }
```

```
          else
            clock_seq[counter] = 0 ;
        }*/
        num_2 = 0 ;
        i-- ;
        while(edit_y[i]==2)
        {
          num_2++ ;
          i-- ;
        }
        counter++ ;
        if(counter<=s)
        {
          for(j=0;j<=E;j++)
          {
            if(num_2==j)
            {
              clock_seq[counter] = j ;
              break ;
            }
          }
        }
      } while(i>0) ;
      if (counter>s)
        counter = s ;
      // Check if the solution has been found
      ++(*n_path) ;
      num_equal_bits = 0 ;
      for(i=1;i<=counter;i++) {
        if(clock_seq[i] == solution[i])
          num_equal_bits++ ;
      }
      if(num_equal_bits == counter){
        free(edit_x) ;
        free(edit_y) ;
        free(clock_seq) ;
        for(i=0;i<s+1;i++){
          free(branch_stack[i].cell.vp) ;
          free(branch_stack[i].cell.vu) ;
          free(branch_stack[i].cell.ve) ;
          free(branch_stack[i].cell.vj) ;
        }
        free(branch_stack) ;
        return true ;
      }
    }
  }
  // If the branching stack is not empty, then erase
     everything up to
  // the branching point at the top of the branching stack
  if(b_sp>0){
    len_prev -= (branch_stack[b_sp].e+branch_stack[b_sp].s) ;
    e1 = branch_stack[b_sp].e ;
```

```
      s1 = branch_stack[b_sp].s ;
    }
  }while(b_sp>0) ;
  free(edit_x) ;
  free(edit_y) ;
  free(clock_seq) ;
  for(i=0;i<s+1;i++){
    free(branch_stack[i].cell.vp) ;
    free(branch_stack[i].cell.vu) ;
    free(branch_stack[i].cell.ve) ;
    free(branch_stack[i].cell.vj) ;
  }
  free(branch_stack) ;
  return false ;
}
```

# B Raw data from the experiment

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.000000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 12 | 142 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 11 | 423 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 12 | 58 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 6 | 1327 | 5 | 5 | 15 | 21 |
| 6 | 0.000000 | 16 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.000000 | 3 | 1985 | 5 | 4 | 19 | 21 |
| 6 | 0.000000 | 10 | 462 | 5 | 2 | 12 | 21 |
| 6 | 0.000000 | 3 | 3685 | 5 | 5 | 18 | 21 |
| 6 | 0.000000 | 12 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.000000 | 12 | 80 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 6 | 1483 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 12 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 10 | 512 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 11 | 116 | 5 | 3 | 11 | 21 |
| 6 | 0.000000 | 1 | 250 | 5 | 4 | 20 | 21 |
| 6 | 0.000000 | 11 | 373 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 13 | 56 | 5 | 2 | 9 | 21 |
| 6 | 0.000000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 13 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.000000 | 5 | 4149 | 5 | 4 | 17 | 21 |
| 6 | 0.000000 | 13 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 11 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 12 | 126 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 14 | 7 | 5 | 1 | 8 | 21 |
| 6 | 0.000000 | 12 | 165 | 5 | 1 | 10 | 21 |
| 6 | 0.000000 | 10 | 310 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 5 | 2549 | 5 | 4 | 16 | 21 |
| 6 | 0.000000 | 13 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.000000 | 8 | 836 | 5 | 2 | 13 | 21 |
| 6 | 0.000000 | 9 | 685 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.000000 | 12 | 111 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 15 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.000000 | 12 | 209 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 3 | 2232 | 5 | 5 | 18 | 21 |
| 6 | 0.000000 | 6 | 2530 | 5 | 3 | 15 | 21 |
| 6 | 0.000000 | 14 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.000000 | 11 | 194 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 11 | 147 | 5 | 2 | 10 | 21 |
| 6 | 0.000000 | 12 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 13 | 75 | 5 | 1 | 9 | 21 |
| 6 | 0.000000 | 11 | 372 | 5 | 2 | 11 | 21 |
| 6 | 0.000000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.000000 | 10 | 399 | 5 | 2 | 11 | 21 |
| 6 | 0.050000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.050000 | 12 | 142 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 11 | 423 | 5 | 2 | 11 | 21 |
| 6 | 0.050000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.050000 | 12 | 56 | 5 | 1 | 9 | 21 |
| 6 | 0.050000 | 6 | 1327 | 5 | 5 | 15 | 21 |
| 6 | 0.050000 | 14 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.050000 | 3 | 1985 | 5 | 4 | 19 | 21 |
| 6 | 0.050000 | 10 | 462 | 5 | 2 | 12 | 21 |
| 6 | 0.050000 | 3 | 3685 | 5 | 5 | 18 | 21 |
| 6 | 0.050000 | 12 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.050000 | 12 | 80 | 5 | 1 | 9 | 21 |
| 6 | 0.050000 | 6 | 1483 | 5 | 3 | 15 | 21 |
| 6 | 0.050000 | 10 | 126 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 10 | 512 | 5 | 3 | 12 | 21 |
| 6 | 0.050000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.050000 | 9 | 116 | 5 | 3 | 11 | 21 |
| 6 | 0.050000 | 1 | 85 | 5 | 4 | 20 | 21 |
| 6 | 0.050000 | 10 | 367 | 5 | 2 | 11 | 21 |
| 6 | 0.050000 | 13 | 56 | 5 | 2 | 9 | 21 |
| 6 | 0.050000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.050000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.050000 | 12 | 9 | 5 | 2 | 9 | 21 |
| 6 | 0.050000 | 5 | 3498 | 5 | 4 | 17 | 21 |
| 6 | 0.050000 | 13 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.050000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.050000 | 11 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 12 | 126 | 5 | 3 | 10 | 21 |
| 6 | 0.050000 | 14 | 7 | 5 | 1 | 8 | 21 |
| 6 | 0.050000 | 12 | 163 | 5 | 1 | 10 | 21 |
| 6 | 0.050000 | 10 | 310 | 5 | 3 | 12 | 21 |
| 6 | 0.050000 | 5 | 2549 | 5 | 4 | 16 | 21 |
| 6 | 0.050000 | 13 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.050000 | 8 | 836 | 5 | 2 | 13 | 21 |
| 6 | 0.050000 | 9 | 685 | 5 | 3 | 12 | 21 |
| 6 | 0.050000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.050000 | 12 | 136 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 13 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.050000 | 12 | 110 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 3 | 2596 | 5 | 5 | 18 | 21 |
| 6 | 0.050000 | 6 | 2505 | 5 | 3 | 15 | 21 |
| 6 | 0.050000 | 14 | 21 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.050000 | 11 | 194 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 11 | 146 | 5 | 2 | 10 | 21 |
| 6 | 0.050000 | 12 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.050000 | 13 | 75 | 5 | 1 | 9 | 21 |
| 6 | 0.050000 | 11 | 372 | 5 | 2 | 11 | 21 |
| 6 | 0.050000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.050000 | 10 | 399 | 5 | 2 | 11 | 21 |
| 6 | 0.100000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.100000 | 10 | 85 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 11 | 423 | 5 | 2 | 11 | 21 |
| 6 | 0.100000 | 16 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.100000 | 12 | 58 | 5 | 1 | 9 | 21 |
| 6 | 0.100000 | 6 | 1236 | 5 | 5 | 15 | 21 |
| 6 | 0.100000 | 15 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.100000 | 3 | 1985 | 5 | 4 | 19 | 21 |
| 6 | 0.100000 | 10 | 462 | 5 | 2 | 12 | 21 |
| 6 | 0.100000 | 3 | 3685 | 5 | 5 | 18 | 21 |
| 6 | 0.100000 | 12 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 9 | 423 | 5 | 2 | 12 | 21 |
| 6 | 0.100000 | 12 | 70 | 5 | 1 | 9 | 21 |
| 6 | 0.100000 | 6 | 1483 | 5 | 3 | 15 | 21 |
| 6 | 0.100000 | 10 | 122 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 10 | 512 | 5 | 3 | 12 | 21 |
| 6 | 0.100000 | 6 | 2634 | 5 | 3 | 15 | 21 |
| 6 | 0.100000 | 10 | 116 | 5 | 3 | 11 | 21 |
| 6 | 0.100000 | 1 | 250 | 5 | 4 | 20 | 21 |
| 6 | 0.100000 | 11 | 373 | 5 | 2 | 11 | 21 |
| 6 | 0.100000 | 13 | 56 | 5 | 2 | 9 | 21 |
| 6 | 0.100000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.100000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.100000 | 13 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.100000 | 5 | 4634 | 5 | 4 | 17 | 21 |
| 6 | 0.100000 | 13 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.100000 | 13 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.100000 | 11 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 12 | 126 | 5 | 3 | 10 | 21 |
| 6 | 0.100000 | 14 | 7 | 5 | 1 | 8 | 21 |
| 6 | 0.100000 | 12 | 162 | 5 | 1 | 10 | 21 |
| 6 | 0.100000 | 10 | 310 | 5 | 3 | 12 | 21 |
| 6 | 0.100000 | 5 | 2549 | 5 | 4 | 16 | 21 |
| 6 | 0.100000 | 11 | 107 | 5 | 3 | 10 | 21 |
| 6 | 0.100000 | 8 | 836 | 5 | 2 | 13 | 21 |
| 6 | 0.100000 | 9 | 810 | 5 | 3 | 12 | 21 |
| 6 | 0.100000 | 7 | 1458 | 5 | 3 | 14 | 21 |
| 6 | 0.100000 | 10 | 55 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 15 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.100000 | 12 | 209 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 3 | 2232 | 5 | 5 | 18 | 21 |
| 6 | 0.100000 | 6 | 1600 | 5 | 3 | 15 | 21 |
| 6 | 0.100000 | 14 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.100000 | 11 | 194 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 11 | 148 | 5 | 2 | 10 | 21 |
| 6 | 0.100000 | 12 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.100000 | 13 | 70 | 5 | 1 | 9 | 21 |
| 6 | 0.100000 | 9 | 225 | 5 | 2 | 11 | 21 |
| 6 | 0.100000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.100000 | 10 | 399 | 5 | 2 | 11 | 21 |
| 6 | 0.150000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.150000 | 12 | 142 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 9 | 367 | 5 | 2 | 11 | 21 |
| 6 | 0.150000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.150000 | 12 | 58 | 5 | 1 | 9 | 21 |
| 6 | 0.150000 | 6 | 106 | 5 | 5 | 15 | 21 |
| 6 | 0.150000 | 15 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.150000 | 3 | 1985 | 5 | 4 | 19 | 21 |
| 6 | 0.150000 | 8 | 210 | 5 | 2 | 12 | 21 |
| 6 | 0.150000 | 3 | 3393 | 5 | 5 | 18 | 21 |
| 6 | 0.150000 | 10 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 9 | 174 | 5 | 2 | 12 | 21 |
| 6 | 0.150000 | 12 | 80 | 5 | 1 | 9 | 21 |
| 6 | 0.150000 | 6 | 858 | 5 | 3 | 15 | 21 |
| 6 | 0.150000 | 12 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 10 | 492 | 5 | 3 | 12 | 21 |
| 6 | 0.150000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.150000 | 11 | 9 | 5 | 3 | 11 | 21 |
| 6 | 0.150000 | 1 | 24 | 5 | 4 | 20 | 21 |
| 6 | 0.150000 | 9 | 244 | 5 | 2 | 11 | 21 |
| 6 | 0.150000 | 12 | 44 | 5 | 2 | 9 | 21 |
| 6 | 0.150000 | 15 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.150000 | 6 | 1333 | 5 | 3 | 15 | 21 |
| 6 | 0.150000 | 12 | 9 | 5 | 2 | 9 | 21 |
| 6 | 0.150000 | 5 | 2568 | 5 | 4 | 17 | 21 |
| 6 | 0.150000 | 13 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.150000 | 13 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.150000 | 11 | 179 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 11 | 2 | 5 | 3 | 10 | 21 |
| 6 | 0.150000 | 14 | 7 | 5 | 1 | 8 | 21 |
| 6 | 0.150000 | 12 | 165 | 5 | 1 | 10 | 21 |
| 6 | 0.150000 | 10 | 308 | 5 | 3 | 12 | 21 |
| 6 | 0.150000 | 5 | 2516 | 5 | 4 | 16 | 21 |
| 6 | 0.150000 | 13 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.150000 | 8 | 826 | 5 | 2 | 13 | 21 |
| 6 | 0.150000 | 9 | 675 | 5 | 3 | 12 | 21 |
| 6 | 0.150000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.150000 | 12 | 111 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 14 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.150000 | 12 | 209 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 3 | 523 | 5 | 5 | 18 | 21 |
| 6 | 0.150000 | 6 | 2530 | 5 | 3 | 15 | 21 |
| 6 | 0.150000 | 14 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.150000 | 11 | 166 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 10 | 141 | 5 | 2 | 10 | 21 |
| 6 | 0.150000 | 11 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.150000 | 12 | 49 | 5 | 1 | 9 | 21 |
| 6 | 0.150000 | 11 | 331 | 5 | 2 | 11 | 21 |
| 6 | 0.150000 | 10 | 364 | 5 | 3 | 12 | 21 |
| 6 | 0.150000 | 10 | 399 | 5 | 2 | 11 | 21 |
| 6 | 0.200000 | 11 | 115 | 5 | 3 | 10 | 21 |
| 6 | 0.200000 | 12 | 72 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 11 | 423 | 5 | 2 | 11 | 21 |
| 6 | 0.200000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.200000 | 12 | 51 | 5 | 1 | 9 | 21 |
| 6 | 0.200000 | 6 | 106 | 5 | 5 | 15 | 21 |
| 6 | 0.200000 | 16 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.200000 | 3 | 1402 | 5 | 4 | 19 | 21 |
| 6 | 0.200000 | 10 | 412 | 5 | 2 | 12 | 21 |
| 6 | 0.200000 | 3 | 3996 | 5 | 5 | 18 | 21 |
| 6 | 0.200000 | 11 | 164 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 9 | 410 | 5 | 2 | 12 | 21 |
| 6 | 0.200000 | 11 | 70 | 5 | 1 | 9 | 21 |
| 6 | 0.200000 | 6 | 1483 | 5 | 3 | 15 | 21 |
| 6 | 0.200000 | 12 | 164 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 8 | 47 | 5 | 3 | 12 | 21 |
| 6 | 0.200000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.200000 | 11 | 116 | 5 | 3 | 11 | 21 |
| 6 | 0.200000 | 1 | 462 | 5 | 4 | 20 | 21 |
| 6 | 0.200000 | 11 | 373 | 5 | 2 | 11 | 21 |
| 6 | 0.200000 | 12 | 46 | 5 | 2 | 9 | 21 |
| 6 | 0.200000 | 14 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.200000 | 6 | 1599 | 5 | 3 | 15 | 21 |
| 6 | 0.200000 | 13 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.200000 | 5 | 4149 | 5 | 4 | 17 | 21 |
| 6 | 0.200000 | 11 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.200000 | 13 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.200000 | 11 | 183 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 11 | 126 | 5 | 3 | 10 | 21 |
| 6 | 0.200000 | 13 | 6 | 5 | 1 | 8 | 21 |
| 6 | 0.200000 | 12 | 160 | 5 | 1 | 10 | 21 |
| 6 | 0.200000 | 10 | 156 | 5 | 3 | 12 | 21 |
| 6 | 0.200000 | 5 | 2509 | 5 | 4 | 16 | 21 |
| 6 | 0.200000 | 9 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.200000 | 8 | 826 | 5 | 2 | 13 | 21 |
| 6 | 0.200000 | 9 | 685 | 5 | 3 | 12 | 21 |
| 6 | 0.200000 | 7 | 726 | 5 | 3 | 14 | 21 |
| 6 | 0.200000 | 11 | 114 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 13 | 4 | 5 | 1 | 7 | 21 |
| 6 | 0.200000 | 12 | 108 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 3 | 2157 | 5 | 5 | 18 | 21 |
| 6 | 0.200000 | 6 | 2505 | 5 | 3 | 15 | 21 |
| 6 | 0.200000 | 10 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.200000 | 11 | 151 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 11 | 143 | 5 | 2 | 10 | 21 |
| 6 | 0.200000 | 11 | 42 | 5 | 1 | 9 | 21 |
| 6 | 0.200000 | 13 | 61 | 5 | 1 | 9 | 21 |
| 6 | 0.200000 | 11 | 352 | 5 | 2 | 11 | 21 |
| 6 | 0.200000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.200000 | 10 | 319 | 5 | 2 | 11 | 21 |
| 6 | 0.250000 | 10 | 58 | 5 | 3 | 10 | 21 |
| 6 | 0.250000 | 11 | 66 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 11 | 419 | 5 | 2 | 11 | 21 |
| 6 | 0.250000 | 11 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.250000 | 11 | 9 | 5 | 1 | 9 | 21 |
| 6 | 0.250000 | 6 | 1236 | 5 | 5 | 15 | 21 |
| 6 | 0.250000 | 16 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.250000 | 3 | 1536 | 5 | 4 | 19 | 21 |
| 6 | 0.250000 | 10 | 389 | 5 | 2 | 12 | 21 |
| 6 | 0.250000 | 3 | 1332 | 5 | 5 | 18 | 21 |
| 6 | 0.250000 | 9 | 166 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.250000 | 12 | 69 | 5 | 1 | 9 | 21 |
| 6 | 0.250000 | 6 | 1222 | 5 | 3 | 15 | 21 |
| 6 | 0.250000 | 12 | 108 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 10 | 512 | 5 | 3 | 12 | 21 |
| 6 | 0.250000 | 6 | 2682 | 5 | 3 | 15 | 21 |
| 6 | 0.250000 | 11 | 115 | 5 | 3 | 11 | 21 |
| 6 | 0.250000 | 1 | 462 | 5 | 4 | 20 | 21 |
| 6 | 0.250000 | 10 | 358 | 5 | 2 | 11 | 21 |
| 6 | 0.250000 | 10 | 34 | 5 | 2 | 9 | 21 |
| 6 | 0.250000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.250000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.250000 | 12 | 9 | 5 | 2 | 9 | 21 |
| 6 | 0.250000 | 5 | 2568 | 5 | 4 | 17 | 21 |
| 6 | 0.250000 | 13 | 2 | 5 | 3 | 10 | 21 |
| 6 | 0.250000 | 13 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.250000 | 11 | 132 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 10 | 5 | 5 | 3 | 10 | 21 |
| 6 | 0.250000 | 12 | 5 | 5 | 1 | 8 | 21 |
| 6 | 0.250000 | 12 | 123 | 5 | 1 | 10 | 21 |
| 6 | 0.250000 | 10 | 311 | 5 | 3 | 12 | 21 |
| 6 | 0.250000 | 5 | 2509 | 5 | 4 | 16 | 21 |
| 6 | 0.250000 | 13 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.250000 | 8 | 836 | 5 | 2 | 13 | 21 |
| 6 | 0.250000 | 9 | 675 | 5 | 3 | 12 | 21 |
| 6 | 0.250000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.250000 | 12 | 136 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 12 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.250000 | 10 | 171 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 3 | 2130 | 5 | 5 | 18 | 21 |
| 6 | 0.250000 | 6 | 1369 | 5 | 3 | 15 | 21 |
| 6 | 0.250000 | 14 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.250000 | 11 | 138 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 11 | 73 | 5 | 2 | 10 | 21 |
| 6 | 0.250000 | 12 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.250000 | 12 | 75 | 5 | 1 | 9 | 21 |
| 6 | 0.250000 | 9 | 100 | 5 | 2 | 11 | 21 |
| 6 | 0.250000 | 10 | 579 | 5 | 3 | 12 | 21 |
| 6 | 0.250000 | 10 | 283 | 5 | 2 | 11 | 21 |
| 6 | 0.300000 | 12 | 144 | 5 | 3 | 10 | 21 |
| 6 | 0.300000 | 9 | 85 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 9 | 363 | 5 | 2 | 11 | 21 |
| 6 | 0.300000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.300000 | 12 | 10 | 5 | 1 | 9 | 21 |
| 6 | 0.300000 | 6 | 106 | 5 | 5 | 15 | 21 |
| 6 | 0.300000 | 10 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.300000 | 3 | 1609 | 5 | 4 | 19 | 21 |
| 6 | 0.300000 | 10 | 401 | 5 | 2 | 12 | 21 |
| 6 | 0.300000 | 3 | 3393 | 5 | 5 | 18 | 21 |
| 6 | 0.300000 | 10 | 111 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.300000 | 12 | 42 | 5 | 1 | 9 | 21 |
| 6 | 0.300000 | 6 | 1479 | 5 | 3 | 15 | 21 |
| 6 | 0.300000 | 12 | 178 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 8 | 237 | 5 | 3 | 12 | 21 |
| 6 | 0.300000 | 6 | 2291 | 5 | 3 | 15 | 21 |
| 6 | 0.300000 | 10 | 88 | 5 | 3 | 11 | 21 |
| 6 | 0.300000 | 1 | 250 | 5 | 4 | 20 | 21 |
| 6 | 0.300000 | 11 | 342 | 5 | 2 | 11 | 21 |
| 6 | 0.300000 | 13 | 46 | 5 | 2 | 9 | 21 |
| 6 | 0.300000 | 13 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.300000 | 6 | 1602 | 5 | 3 | 15 | 21 |
| 6 | 0.300000 | 11 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.300000 | 5 | 4659 | 5 | 4 | 17 | 21 |
| 6 | 0.300000 | 9 | 58 | 5 | 3 | 10 | 21 |
| 6 | 0.300000 | 12 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.300000 | 12 | 132 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 10 | 5 | 5 | 3 | 10 | 21 |
| 6 | 0.300000 | 14 | 4 | 5 | 1 | 8 | 21 |
| 6 | 0.300000 | 12 | 162 | 5 | 1 | 10 | 21 |
| 6 | 0.300000 | 7 | 30 | 5 | 3 | 12 | 21 |
| 6 | 0.300000 | 5 | 2549 | 5 | 4 | 16 | 21 |
| 6 | 0.300000 | 11 | 113 | 5 | 3 | 10 | 21 |
| 6 | 0.300000 | 8 | 736 | 5 | 2 | 13 | 21 |
| 6 | 0.300000 | 9 | 685 | 5 | 3 | 12 | 21 |
| 6 | 0.300000 | 7 | 724 | 5 | 3 | 14 | 21 |
| 6 | 0.300000 | 12 | 136 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 15 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.300000 | 12 | 209 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 3 | 2130 | 5 | 5 | 18 | 21 |
| 6 | 0.300000 | 6 | 1327 | 5 | 3 | 15 | 21 |
| 6 | 0.300000 | 14 | 13 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.300000 | 11 | 135 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 11 | 148 | 5 | 2 | 10 | 21 |
| 6 | 0.300000 | 11 | 37 | 5 | 1 | 9 | 21 |
| 6 | 0.300000 | 12 | 70 | 5 | 1 | 9 | 21 |
| 6 | 0.300000 | 10 | 242 | 5 | 2 | 11 | 21 |
| 6 | 0.300000 | 10 | 365 | 5 | 3 | 12 | 21 |
| 6 | 0.300000 | 9 | 196 | 5 | 2 | 11 | 21 |
| 6 | 0.350000 | 10 | 58 | 5 | 3 | 10 | 21 |
| 6 | 0.350000 | 11 | 140 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 11 | 75 | 5 | 2 | 11 | 21 |
| 6 | 0.350000 | 14 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.350000 | 12 | 56 | 5 | 1 | 9 | 21 |
| 6 | 0.350000 | 6 | 106 | 5 | 5 | 15 | 21 |
| 6 | 0.350000 | 12 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.350000 | 3 | 5539 | 5 | 4 | 19 | 21 |
| 6 | 0.350000 | 10 | 521 | 5 | 2 | 12 | 21 |
| 6 | 0.350000 | 3 | 283 | 5 | 5 | 18 | 21 |
| 6 | 0.350000 | 12 | 167 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 9 | 429 | 5 | 2 | 12 | 21 |
| 6 | 0.350000 | 12 | 53 | 5 | 1 | 9 | 21 |
| 6 | 0.350000 | 6 | 858 | 5 | 3 | 15 | 21 |
| 6 | 0.350000 | 10 | 118 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 10 | 511 | 5 | 3 | 12 | 21 |
| 6 | 0.350000 | 6 | 1521 | 5 | 3 | 15 | 21 |
| 6 | 0.350000 | 11 | 193 | 5 | 3 | 11 | 21 |
| 6 | 0.350000 | 1 | 213 | 5 | 4 | 20 | 21 |
| 6 | 0.350000 | 11 | 367 | 5 | 2 | 11 | 21 |
| 6 | 0.350000 | 11 | 45 | 5 | 2 | 9 | 21 |
| 6 | 0.350000 | 15 | 3 | 5 | 1 | 7 | 21 |
| 6 | 0.350000 | 6 | 1633 | 5 | 3 | 15 | 21 |
| 6 | 0.350000 | 13 | 18 | 5 | 2 | 9 | 21 |
| 6 | 0.350000 | 6 | 2543 | 5 | 4 | 17 | 21 |
| 6 | 0.350000 | 11 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.350000 | 12 | 4 | 5 | 1 | 7 | 21 |
| 6 | 0.350000 | 10 | 69 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 11 | 17 | 5 | 3 | 10 | 21 |
| 6 | 0.350000 | 13 | 5 | 5 | 1 | 8 | 21 |
| 6 | 0.350000 | 10 | 67 | 5 | 1 | 10 | 21 |
| 6 | 0.350000 | 10 | 32 | 5 | 3 | 12 | 21 |
| 6 | 0.350000 | 5 | 797 | 5 | 4 | 16 | 21 |
| 6 | 0.350000 | 11 | 58 | 5 | 3 | 10 | 21 |
| 6 | 0.350000 | 8 | 370 | 5 | 2 | 13 | 21 |
| 6 | 0.350000 | 9 | 320 | 5 | 3 | 12 | 21 |
| 6 | 0.350000 | 7 | 1268 | 5 | 3 | 14 | 21 |
| 6 | 0.350000 | 12 | 86 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 14 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.350000 | 10 | 171 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 3 | 2445 | 5 | 5 | 18 | 21 |
| 6 | 0.350000 | 6 | 1173 | 5 | 3 | 15 | 21 |
| 6 | 0.350000 | 13 | 13 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.350000 | 11 | 155 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 11 | 141 | 5 | 2 | 10 | 21 |
| 6 | 0.350000 | 11 | 43 | 5 | 1 | 9 | 21 |
| 6 | 0.350000 | 13 | 75 | 5 | 1 | 9 | 21 |
| 6 | 0.350000 | 9 | 176 | 5 | 2 | 11 | 21 |
| 6 | 0.350000 | 10 | 564 | 5 | 3 | 12 | 21 |
| 6 | 0.350000 | 10 | 183 | 5 | 2 | 11 | 21 |
| 6 | 0.400000 | 12 | 109 | 5 | 3 | 10 | 21 |
| 6 | 0.400000 | 9 | 7 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 11 | 102 | 5 | 2 | 11 | 21 |
| 6 | 0.400000 | 13 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.400000 | 11 | 56 | 5 | 1 | 9 | 21 |
| 6 | 0.400000 | 6 | 1236 | 5 | 5 | 15 | 21 |
| 6 | 0.400000 | 16 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.400000 | 3 | 932 | 5 | 4 | 19 | 21 |
| 6 | 0.400000 | 10 | 461 | 5 | 2 | 12 | 21 |
| 6 | 0.400000 | 3 | 1165 | 5 | 5 | 18 | 21 |
| 6 | 0.400000 | 10 | 110 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 9 | 154 | 5 | 2 | 12 | 21 |
| 6 | 0.400000 | 12 | 52 | 5 | 1 | 9 | 21 |
| 6 | 0.400000 | 6 | 1274 | 5 | 3 | 15 | 21 |
| 6 | 0.400000 | 10 | 122 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 10 | 284 | 5 | 3 | 12 | 21 |
| 6 | 0.400000 | 6 | 2453 | 5 | 3 | 15 | 21 |
| 6 | 0.400000 | 11 | 191 | 5 | 3 | 11 | 21 |
| 6 | 0.400000 | 1 | 250 | 5 | 4 | 20 | 21 |
| 6 | 0.400000 | 11 | 367 | 5 | 2 | 11 | 21 |
| 6 | 0.400000 | 10 | 23 | 5 | 2 | 9 | 21 |
| 6 | 0.400000 | 12 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.400000 | 6 | 862 | 5 | 3 | 15 | 21 |
| 6 | 0.400000 | 11 | 3 | 5 | 2 | 9 | 21 |
| 6 | 0.400000 | 5 | 2568 | 5 | 4 | 17 | 21 |
| 6 | 0.400000 | 11 | 86 | 5 | 3 | 10 | 21 |
| 6 | 0.400000 | 12 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.400000 | 11 | 200 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 9 | 2 | 5 | 3 | 10 | 21 |
| 6 | 0.400000 | 14 | 5 | 5 | 1 | 8 | 21 |
| 6 | 0.400000 | 10 | 67 | 5 | 1 | 10 | 21 |
| 6 | 0.400000 | 8 | 32 | 5 | 3 | 12 | 21 |
| 6 | 0.400000 | 5 | 2542 | 5 | 4 | 16 | 21 |
| 6 | 0.400000 | 10 | 78 | 5 | 3 | 10 | 21 |
| 6 | 0.400000 | 7 | 746 | 5 | 2 | 13 | 21 |
| 6 | 0.400000 | 9 | 287 | 5 | 3 | 12 | 21 |
| 6 | 0.400000 | 7 | 632 | 5 | 3 | 14 | 21 |
| 6 | 0.400000 | 10 | 55 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 13 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.400000 | 10 | 102 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 3 | 276 | 5 | 5 | 18 | 21 |
| 6 | 0.400000 | 6 | 1558 | 5 | 3 | 15 | 21 |
| 6 | 0.400000 | 13 | 14 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.400000 | 10 | 96 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 10 | 71 | 5 | 2 | 10 | 21 |
| 6 | 0.400000 | 11 | 21 | 5 | 1 | 9 | 21 |
| 6 | 0.400000 | 13 | 39 | 5 | 1 | 9 | 21 |
| 6 | 0.400000 | 9 | 219 | 5 | 2 | 11 | 21 |
| 6 | 0.400000 | 10 | 578 | 5 | 3 | 12 | 21 |
| 6 | 0.400000 | 10 | 186 | 5 | 2 | 11 | 21 |
| 6 | 0.450000 | 12 | 114 | 5 | 3 | 10 | 21 |
| 6 | 0.450000 | 11 | 141 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 9 | 140 | 5 | 2 | 11 | 21 |
| 6 | 0.450000 | 11 | 6 | 5 | 1 | 7 | 21 |
| 6 | 0.450000 | 12 | 10 | 5 | 1 | 9 | 21 |
| 6 | 0.450000 | 6 | 1236 | 5 | 5 | 15 | 21 |
| 6 | 0.450000 | 14 | 1 | 5 | 0 | 6 | 21 |
| 6 | 0.450000 | 3 | 1923 | 5 | 4 | 19 | 21 |
| 6 | 0.450000 | 9 | 395 | 5 | 2 | 12 | 21 |
| 6 | 0.450000 | 4 | 3208 | 5 | 5 | 18 | 21 |
| 6 | 0.450000 | 9 | 164 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 9 | 168 | 5 | 2 | 12 | 21 |
| 6 | 0.450000 | 10 | 64 | 5 | 1 | 9 | 21 |
| 6 | 0.450000 | 6 | 594 | 5 | 3 | 15 | 21 |
| 6 | 0.450000 | 12 | 175 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 9 | 509 | 5 | 3 | 12 | 21 |
| 6 | 0.450000 | 6 | 1146 | 5 | 3 | 15 | 21 |
| 6 | 0.450000 | 9 | 115 | 5 | 3 | 11 | 21 |
| 6 | 0.450000 | 3 | 3598 | 5 | 4 | 20 | 21 |
| 6 | 0.450000 | 11 | 342 | 5 | 2 | 11 | 21 |
| 6 | 0.450000 | 10 | 24 | 5 | 2 | 9 | 21 |
| 6 | 0.450000 | 11 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.450000 | 6 | 874 | 5 | 3 | 15 | 21 |
| 6 | 0.450000 | 11 | 9 | 5 | 2 | 9 | 21 |
| 6 | 0.450000 | 5 | 4634 | 5 | 4 | 17 | 21 |
| 6 | 0.450000 | 11 | 122 | 5 | 3 | 10 | 21 |
| 6 | 0.450000 | 13 | 7 | 5 | 1 | 7 | 21 |
| 6 | 0.450000 | 10 | 172 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 12 | 2 | 5 | 3 | 10 | 21 |
| 6 | 0.450000 | 13 | 6 | 5 | 1 | 8 | 21 |
| 6 | 0.450000 | 10 | 81 | 5 | 1 | 10 | 21 |
| 6 | 0.450000 | 8 | 35 | 5 | 3 | 12 | 21 |
| 6 | 0.450000 | 5 | 797 | 5 | 4 | 16 | 21 |
| 6 | 0.450000 | 10 | 72 | 5 | 3 | 10 | 21 |
| 6 | 0.450000 | 8 | 826 | 5 | 2 | 13 | 21 |
| 6 | 0.450000 | 9 | 319 | 5 | 3 | 12 | 21 |
| 6 | 0.450000 | 7 | 1482 | 5 | 3 | 14 | 21 |
| 6 | 0.450000 | 10 | 53 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 12 | 5 | 5 | 1 | 7 | 21 |
| 6 | 0.450000 | 12 | 201 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 3 | 2471 | 5 | 5 | 18 | 21 |
| 6 | 0.450000 | 6 | 2514 | 5 | 3 | 15 | 21 |
| 6 | 0.450000 | 13 | 13 | 5 | 1 | 8 | 21 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 6 | 0.450000 | 11 | 128 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 11 | 76 | 5 | 2 | 10 | 21 |
| 6 | 0.450000 | 11 | 21 | 5 | 1 | 9 | 21 |
| 6 | 0.450000 | 9 | 33 | 5 | 1 | 9 | 21 |
| 6 | 0.450000 | 9 | 226 | 5 | 2 | 11 | 21 |
| 6 | 0.450000 | 10 | 570 | 5 | 3 | 12 | 21 |
| 6 | 0.450000 | 10 | 173 | 5 | 2 | 11 | 21 |
| 7 | 0.000000 | 17 | 84 | 6 | 4 | 12 | 28 |
| 7 | 0.000000 | 15 | 881 | 6 | 2 | 13 | 28 |
| 7 | 0.000000 | 17 | 172 | 6 | 3 | 12 | 28 |
| 7 | 0.000000 | 8 | 19568 | 6 | 6 | 21 | 28 |
| 7 | 0.000000 | 15 | 280 | 6 | 2 | 13 | 28 |
| 7 | 0.000000 | 18 | 69 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 16 | 819 | 6 | 3 | 13 | 28 |
| 7 | 0.000000 | 3 | 64214 | 6 | 6 | 25 | 28 |
| 7 | 0.000000 | 15 | 838 | 6 | 4 | 13 | 28 |
| 7 | 0.000000 | 6 | 60137 | 6 | 6 | 22 | 28 |
| 7 | 0.000000 | 13 | 4242 | 6 | 4 | 15 | 28 |
| 7 | 0.000000 | 17 | 448 | 6 | 2 | 12 | 28 |
| 7 | 0.000000 | 5 | 25867 | 6 | 6 | 23 | 28 |
| 7 | 0.000000 | 17 | 291 | 6 | 2 | 11 | 28 |
| 7 | 0.000000 | 15 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.000000 | 18 | 288 | 6 | 4 | 11 | 28 |
| 7 | 0.000000 | 13 | 4041 | 6 | 3 | 15 | 28 |
| 7 | 0.000000 | 16 | 620 | 6 | 3 | 12 | 28 |
| 7 | 0.000000 | 14 | 1442 | 6 | 2 | 14 | 28 |
| 7 | 0.000000 | 15 | 888 | 6 | 3 | 14 | 28 |
| 7 | 0.000000 | 19 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.000000 | 18 | 102 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 19 | 23 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 20 | 23 | 6 | 1 | 9 | 28 |
| 7 | 0.000000 | 15 | 1590 | 6 | 4 | 14 | 28 |
| 7 | 0.000000 | 5 | 73125 | 6 | 6 | 23 | 28 |
| 7 | 0.000000 | 18 | 112 | 6 | 2 | 10 | 28 |
| 7 | 0.000000 | 17 | 112 | 6 | 2 | 11 | 28 |
| 7 | 0.000000 | 4 | 45596 | 6 | 6 | 25 | 28 |
| 7 | 0.000000 | 15 | 1315 | 6 | 2 | 13 | 28 |
| 7 | 0.000000 | 18 | 91 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 17 | 452 | 6 | 2 | 12 | 28 |
| 7 | 0.000000 | 18 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.000000 | 17 | 721 | 6 | 2 | 12 | 28 |
| 7 | 0.000000 | 18 | 108 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 18 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.000000 | 14 | 4511 | 6 | 4 | 15 | 28 |
| 7 | 0.000000 | 18 | 198 | 6 | 1 | 11 | 28 |
| 7 | 0.000000 | 18 | 284 | 6 | 2 | 11 | 28 |
| 7 | 0.000000 | 16 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.000000 | 19 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.000000 | 17 | 294 | 6 | 1 | 11 | 28 |
| 7 | 0.000000 | 7 | 59670 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.000000 | 19 | 112 | 6 | 1 | 10 | 28 |
| 7 | 0.000000 | 16 | 774 | 6 | 2 | 12 | 28 |
| 7 | 0.000000 | 18 | 84 | 6 | 2 | 10 | 28 |
| 7 | 0.000000 | 18 | 307 | 6 | 2 | 11 | 28 |
| 7 | 0.000000 | 13 | 4652 | 6 | 4 | 15 | 28 |
| 7 | 0.000000 | 16 | 1384 | 6 | 4 | 13 | 28 |
| 7 | 0.000000 | 16 | 458 | 6 | 2 | 12 | 28 |
| 7 | 0.050000 | 17 | 84 | 6 | 4 | 12 | 28 |
| 7 | 0.050000 | 15 | 881 | 6 | 2 | 13 | 28 |
| 7 | 0.050000 | 17 | 172 | 6 | 3 | 12 | 28 |
| 7 | 0.050000 | 8 | 19552 | 6 | 6 | 21 | 28 |
| 7 | 0.050000 | 15 | 280 | 6 | 2 | 13 | 28 |
| 7 | 0.050000 | 18 | 69 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 14 | 819 | 6 | 3 | 13 | 28 |
| 7 | 0.050000 | 3 | 64214 | 6 | 6 | 25 | 28 |
| 7 | 0.050000 | 15 | 838 | 6 | 4 | 13 | 28 |
| 7 | 0.050000 | 6 | 60137 | 6 | 6 | 22 | 28 |
| 7 | 0.050000 | 13 | 3891 | 6 | 4 | 15 | 28 |
| 7 | 0.050000 | 17 | 448 | 6 | 2 | 12 | 28 |
| 7 | 0.050000 | 5 | 24362 | 6 | 6 | 23 | 28 |
| 7 | 0.050000 | 17 | 291 | 6 | 2 | 11 | 28 |
| 7 | 0.050000 | 14 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.050000 | 18 | 288 | 6 | 4 | 11 | 28 |
| 7 | 0.050000 | 13 | 4041 | 6 | 3 | 15 | 28 |
| 7 | 0.050000 | 16 | 458 | 6 | 3 | 12 | 28 |
| 7 | 0.050000 | 14 | 1442 | 6 | 2 | 14 | 28 |
| 7 | 0.050000 | 15 | 806 | 6 | 3 | 14 | 28 |
| 7 | 0.050000 | 19 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.050000 | 18 | 102 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 19 | 23 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 20 | 23 | 6 | 1 | 9 | 28 |
| 7 | 0.050000 | 15 | 1590 | 6 | 4 | 14 | 28 |
| 7 | 0.050000 | 5 | 72646 | 6 | 6 | 23 | 28 |
| 7 | 0.050000 | 18 | 97 | 6 | 2 | 10 | 28 |
| 7 | 0.050000 | 16 | 106 | 6 | 2 | 11 | 28 |
| 7 | 0.050000 | 4 | 45596 | 6 | 6 | 25 | 28 |
| 7 | 0.050000 | 15 | 1315 | 6 | 2 | 13 | 28 |
| 7 | 0.050000 | 18 | 91 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 17 | 452 | 6 | 2 | 12 | 28 |
| 7 | 0.050000 | 18 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.050000 | 16 | 538 | 6 | 2 | 12 | 28 |
| 7 | 0.050000 | 17 | 108 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 18 | 197 | 6 | 1 | 11 | 28 |
| 7 | 0.050000 | 14 | 4511 | 6 | 4 | 15 | 28 |
| 7 | 0.050000 | 18 | 198 | 6 | 1 | 11 | 28 |
| 7 | 0.050000 | 18 | 284 | 6 | 2 | 11 | 28 |
| 7 | 0.050000 | 15 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.050000 | 19 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.050000 | 17 | 294 | 6 | 1 | 11 | 28 |
| 7 | 0.050000 | 7 | 59670 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.050000 | 19 | 97 | 6 | 1 | 10 | 28 |
| 7 | 0.050000 | 16 | 774 | 6 | 2 | 12 | 28 |
| 7 | 0.050000 | 18 | 84 | 6 | 2 | 10 | 28 |
| 7 | 0.050000 | 16 | 161 | 6 | 2 | 11 | 28 |
| 7 | 0.050000 | 13 | 4641 | 6 | 4 | 15 | 28 |
| 7 | 0.050000 | 16 | 1384 | 6 | 4 | 13 | 28 |
| 7 | 0.050000 | 15 | 423 | 6 | 2 | 12 | 28 |
| 7 | 0.100000 | 17 | 84 | 6 | 4 | 12 | 28 |
| 7 | 0.100000 | 15 | 881 | 6 | 2 | 13 | 28 |
| 7 | 0.100000 | 16 | 87 | 6 | 3 | 12 | 28 |
| 7 | 0.100000 | 8 | 12829 | 6 | 6 | 21 | 28 |
| 7 | 0.100000 | 15 | 71 | 6 | 2 | 13 | 28 |
| 7 | 0.100000 | 18 | 41 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 16 | 819 | 6 | 3 | 13 | 28 |
| 7 | 0.100000 | 3 | 60796 | 6 | 6 | 25 | 28 |
| 7 | 0.100000 | 15 | 838 | 6 | 4 | 13 | 28 |
| 7 | 0.100000 | 6 | 60137 | 6 | 6 | 22 | 28 |
| 7 | 0.100000 | 13 | 2473 | 6 | 4 | 15 | 28 |
| 7 | 0.100000 | 17 | 310 | 6 | 2 | 12 | 28 |
| 7 | 0.100000 | 5 | 25867 | 6 | 6 | 23 | 28 |
| 7 | 0.100000 | 17 | 291 | 6 | 2 | 11 | 28 |
| 7 | 0.100000 | 15 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.100000 | 18 | 288 | 6 | 4 | 11 | 28 |
| 7 | 0.100000 | 13 | 4041 | 6 | 3 | 15 | 28 |
| 7 | 0.100000 | 16 | 458 | 6 | 3 | 12 | 28 |
| 7 | 0.100000 | 14 | 981 | 6 | 2 | 14 | 28 |
| 7 | 0.100000 | 15 | 888 | 6 | 3 | 14 | 28 |
| 7 | 0.100000 | 19 | 26 | 6 | 1 | 9 | 28 |
| 7 | 0.100000 | 18 | 100 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 19 | 23 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 20 | 23 | 6 | 1 | 9 | 28 |
| 7 | 0.100000 | 15 | 1571 | 6 | 4 | 14 | 28 |
| 7 | 0.100000 | 5 | 73125 | 6 | 6 | 23 | 28 |
| 7 | 0.100000 | 18 | 55 | 6 | 2 | 10 | 28 |
| 7 | 0.100000 | 17 | 112 | 6 | 2 | 11 | 28 |
| 7 | 0.100000 | 4 | 45596 | 6 | 6 | 25 | 28 |
| 7 | 0.100000 | 15 | 1315 | 6 | 2 | 13 | 28 |
| 7 | 0.100000 | 18 | 91 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 17 | 452 | 6 | 2 | 12 | 28 |
| 7 | 0.100000 | 17 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.100000 | 17 | 704 | 6 | 2 | 12 | 28 |
| 7 | 0.100000 | 18 | 108 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 18 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.100000 | 14 | 3786 | 6 | 4 | 15 | 28 |
| 7 | 0.100000 | 18 | 198 | 6 | 1 | 11 | 28 |
| 7 | 0.100000 | 18 | 200 | 6 | 2 | 11 | 28 |
| 7 | 0.100000 | 15 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.100000 | 18 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.100000 | 17 | 294 | 6 | 1 | 11 | 28 |
| 7 | 0.100000 | 7 | 59670 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.100000 | 18 | 97 | 6 | 1 | 10 | 28 |
| 7 | 0.100000 | 16 | 711 | 6 | 2 | 12 | 28 |
| 7 | 0.100000 | 18 | 84 | 6 | 2 | 10 | 28 |
| 7 | 0.100000 | 17 | 307 | 6 | 2 | 11 | 28 |
| 7 | 0.100000 | 13 | 4652 | 6 | 4 | 15 | 28 |
| 7 | 0.100000 | 16 | 1132 | 6 | 4 | 13 | 28 |
| 7 | 0.100000 | 16 | 458 | 6 | 2 | 12 | 28 |
| 7 | 0.150000 | 17 | 84 | 6 | 4 | 12 | 28 |
| 7 | 0.150000 | 15 | 1441 | 6 | 2 | 13 | 28 |
| 7 | 0.150000 | 17 | 172 | 6 | 3 | 12 | 28 |
| 7 | 0.150000 | 8 | 19568 | 6 | 6 | 21 | 28 |
| 7 | 0.150000 | 14 | 279 | 6 | 2 | 13 | 28 |
| 7 | 0.150000 | 18 | 68 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 15 | 819 | 6 | 3 | 13 | 28 |
| 7 | 0.150000 | 3 | 64214 | 6 | 6 | 25 | 28 |
| 7 | 0.150000 | 15 | 836 | 6 | 4 | 13 | 28 |
| 7 | 0.150000 | 6 | 49439 | 6 | 6 | 22 | 28 |
| 7 | 0.150000 | 13 | 764 | 6 | 4 | 15 | 28 |
| 7 | 0.150000 | 15 | 446 | 6 | 2 | 12 | 28 |
| 7 | 0.150000 | 5 | 25867 | 6 | 6 | 23 | 28 |
| 7 | 0.150000 | 17 | 291 | 6 | 2 | 11 | 28 |
| 7 | 0.150000 | 14 | 708 | 6 | 3 | 13 | 28 |
| 7 | 0.150000 | 18 | 120 | 6 | 4 | 11 | 28 |
| 7 | 0.150000 | 13 | 4040 | 6 | 3 | 15 | 28 |
| 7 | 0.150000 | 16 | 620 | 6 | 3 | 12 | 28 |
| 7 | 0.150000 | 14 | 1442 | 6 | 2 | 14 | 28 |
| 7 | 0.150000 | 15 | 888 | 6 | 3 | 14 | 28 |
| 7 | 0.150000 | 18 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.150000 | 17 | 102 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 19 | 23 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 19 | 21 | 6 | 1 | 9 | 28 |
| 7 | 0.150000 | 15 | 1072 | 6 | 4 | 14 | 28 |
| 7 | 0.150000 | 5 | 39803 | 6 | 6 | 23 | 28 |
| 7 | 0.150000 | 18 | 112 | 6 | 2 | 10 | 28 |
| 7 | 0.150000 | 16 | 106 | 6 | 2 | 11 | 28 |
| 7 | 0.150000 | 4 | 47599 | 6 | 6 | 25 | 28 |
| 7 | 0.150000 | 15 | 1063 | 6 | 2 | 13 | 28 |
| 7 | 0.150000 | 17 | 91 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 15 | 452 | 6 | 2 | 12 | 28 |
| 7 | 0.150000 | 17 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.150000 | 17 | 721 | 6 | 2 | 12 | 28 |
| 7 | 0.150000 | 18 | 18 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 18 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.150000 | 14 | 4367 | 6 | 4 | 15 | 28 |
| 7 | 0.150000 | 17 | 187 | 6 | 1 | 11 | 28 |
| 7 | 0.150000 | 18 | 284 | 6 | 2 | 11 | 28 |
| 7 | 0.150000 | 15 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.150000 | 18 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.150000 | 17 | 294 | 6 | 1 | 11 | 28 |
| 7 | 0.150000 | 7 | 59135 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.150000 | 18 | 112 | 6 | 1 | 10 | 28 |
| 7 | 0.150000 | 16 | 416 | 6 | 2 | 12 | 28 |
| 7 | 0.150000 | 18 | 84 | 6 | 2 | 10 | 28 |
| 7 | 0.150000 | 18 | 307 | 6 | 2 | 11 | 28 |
| 7 | 0.150000 | 13 | 3961 | 6 | 4 | 15 | 28 |
| 7 | 0.150000 | 14 | 793 | 6 | 4 | 13 | 28 |
| 7 | 0.150000 | 16 | 458 | 6 | 2 | 12 | 28 |
| 7 | 0.200000 | 17 | 82 | 6 | 4 | 12 | 28 |
| 7 | 0.200000 | 15 | 1399 | 6 | 2 | 13 | 28 |
| 7 | 0.200000 | 17 | 172 | 6 | 3 | 12 | 28 |
| 7 | 0.200000 | 8 | 19321 | 6 | 6 | 21 | 28 |
| 7 | 0.200000 | 15 | 1035 | 6 | 2 | 13 | 28 |
| 7 | 0.200000 | 18 | 68 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 16 | 819 | 6 | 3 | 13 | 28 |
| 7 | 0.200000 | 3 | 46391 | 6 | 6 | 25 | 28 |
| 7 | 0.200000 | 15 | 815 | 6 | 4 | 13 | 28 |
| 7 | 0.200000 | 6 | 49439 | 6 | 6 | 22 | 28 |
| 7 | 0.200000 | 13 | 4242 | 6 | 4 | 15 | 28 |
| 7 | 0.200000 | 15 | 411 | 6 | 2 | 12 | 28 |
| 7 | 0.200000 | 5 | 24594 | 6 | 6 | 23 | 28 |
| 7 | 0.200000 | 17 | 291 | 6 | 2 | 11 | 28 |
| 7 | 0.200000 | 15 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.200000 | 18 | 288 | 6 | 4 | 11 | 28 |
| 7 | 0.200000 | 13 | 4040 | 6 | 3 | 15 | 28 |
| 7 | 0.200000 | 16 | 620 | 6 | 3 | 12 | 28 |
| 7 | 0.200000 | 14 | 1441 | 6 | 2 | 14 | 28 |
| 7 | 0.200000 | 15 | 742 | 6 | 3 | 14 | 28 |
| 7 | 0.200000 | 18 | 6 | 6 | 1 | 9 | 28 |
| 7 | 0.200000 | 18 | 100 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 19 | 21 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 19 | 14 | 6 | 1 | 9 | 28 |
| 7 | 0.200000 | 15 | 1489 | 6 | 4 | 14 | 28 |
| 7 | 0.200000 | 5 | 72802 | 6 | 6 | 23 | 28 |
| 7 | 0.200000 | 18 | 112 | 6 | 2 | 10 | 28 |
| 7 | 0.200000 | 16 | 106 | 6 | 2 | 11 | 28 |
| 7 | 0.200000 | 4 | 20419 | 6 | 6 | 25 | 28 |
| 7 | 0.200000 | 15 | 1113 | 6 | 2 | 13 | 28 |
| 7 | 0.200000 | 18 | 91 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 17 | 452 | 6 | 2 | 12 | 28 |
| 7 | 0.200000 | 17 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.200000 | 17 | 554 | 6 | 2 | 12 | 28 |
| 7 | 0.200000 | 17 | 52 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 16 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.200000 | 14 | 4501 | 6 | 4 | 15 | 28 |
| 7 | 0.200000 | 18 | 198 | 6 | 1 | 11 | 28 |
| 7 | 0.200000 | 17 | 266 | 6 | 2 | 11 | 28 |
| 7 | 0.200000 | 16 | 217 | 6 | 4 | 12 | 28 |
| 7 | 0.200000 | 18 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.200000 | 17 | 294 | 6 | 1 | 11 | 28 |
| 7 | 0.200000 | 7 | 59664 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.200000 | 18 | 41 | 6 | 1 | 10 | 28 |
| 7 | 0.200000 | 16 | 416 | 6 | 2 | 12 | 28 |
| 7 | 0.200000 | 18 | 82 | 6 | 2 | 10 | 28 |
| 7 | 0.200000 | 18 | 305 | 6 | 2 | 11 | 28 |
| 7 | 0.200000 | 13 | 4594 | 6 | 4 | 15 | 28 |
| 7 | 0.200000 | 16 | 1132 | 6 | 4 | 13 | 28 |
| 7 | 0.200000 | 16 | 494 | 6 | 2 | 12 | 28 |
| 7 | 0.250000 | 15 | 81 | 6 | 4 | 12 | 28 |
| 7 | 0.250000 | 15 | 593 | 6 | 2 | 13 | 28 |
| 7 | 0.250000 | 17 | 66 | 6 | 3 | 12 | 28 |
| 7 | 0.250000 | 8 | 19507 | 6 | 6 | 21 | 28 |
| 7 | 0.250000 | 15 | 1035 | 6 | 2 | 13 | 28 |
| 7 | 0.250000 | 18 | 69 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 16 | 687 | 6 | 3 | 13 | 28 |
| 7 | 0.250000 | 3 | 30306 | 6 | 6 | 25 | 28 |
| 7 | 0.250000 | 15 | 793 | 6 | 4 | 13 | 28 |
| 7 | 0.250000 | 6 | 49598 | 6 | 6 | 22 | 28 |
| 7 | 0.250000 | 13 | 3891 | 6 | 4 | 15 | 28 |
| 7 | 0.250000 | 17 | 737 | 6 | 2 | 12 | 28 |
| 7 | 0.250000 | 5 | 25867 | 6 | 6 | 23 | 28 |
| 7 | 0.250000 | 15 | 25 | 6 | 2 | 11 | 28 |
| 7 | 0.250000 | 13 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.250000 | 17 | 204 | 6 | 4 | 11 | 28 |
| 7 | 0.250000 | 13 | 2678 | 6 | 3 | 15 | 28 |
| 7 | 0.250000 | 16 | 325 | 6 | 3 | 12 | 28 |
| 7 | 0.250000 | 14 | 978 | 6 | 2 | 14 | 28 |
| 7 | 0.250000 | 15 | 803 | 6 | 3 | 14 | 28 |
| 7 | 0.250000 | 18 | 6 | 6 | 1 | 9 | 28 |
| 7 | 0.250000 | 18 | 44 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 19 | 21 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 17 | 21 | 6 | 1 | 9 | 28 |
| 7 | 0.250000 | 15 | 1466 | 6 | 4 | 14 | 28 |
| 7 | 0.250000 | 5 | 54094 | 6 | 6 | 23 | 28 |
| 7 | 0.250000 | 18 | 92 | 6 | 2 | 10 | 28 |
| 7 | 0.250000 | 15 | 28 | 6 | 2 | 11 | 28 |
| 7 | 0.250000 | 4 | 45224 | 6 | 6 | 25 | 28 |
| 7 | 0.250000 | 15 | 1208 | 6 | 2 | 13 | 28 |
| 7 | 0.250000 | 18 | 22 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 17 | 413 | 6 | 2 | 12 | 28 |
| 7 | 0.250000 | 18 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.250000 | 16 | 721 | 6 | 2 | 12 | 28 |
| 7 | 0.250000 | 17 | 52 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 16 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.250000 | 14 | 3779 | 6 | 4 | 15 | 28 |
| 7 | 0.250000 | 16 | 190 | 6 | 1 | 11 | 28 |
| 7 | 0.250000 | 18 | 294 | 6 | 2 | 11 | 28 |
| 7 | 0.250000 | 16 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.250000 | 17 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.250000 | 17 | 231 | 6 | 1 | 11 | 28 |
| 7 | 0.250000 | 7 | 59670 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.250000 | 16 | 85 | 6 | 1 | 10 | 28 |
| 7 | 0.250000 | 16 | 711 | 6 | 2 | 12 | 28 |
| 7 | 0.250000 | 18 | 84 | 6 | 2 | 10 | 28 |
| 7 | 0.250000 | 17 | 245 | 6 | 2 | 11 | 28 |
| 7 | 0.250000 | 13 | 2079 | 6 | 4 | 15 | 28 |
| 7 | 0.250000 | 16 | 1384 | 6 | 4 | 13 | 28 |
| 7 | 0.250000 | 15 | 423 | 6 | 2 | 12 | 28 |
| 7 | 0.300000 | 16 | 7 | 6 | 4 | 12 | 28 |
| 7 | 0.300000 | 15 | 448 | 6 | 2 | 13 | 28 |
| 7 | 0.300000 | 16 | 102 | 6 | 3 | 12 | 28 |
| 7 | 0.300000 | 8 | 19549 | 6 | 6 | 21 | 28 |
| 7 | 0.300000 | 15 | 951 | 6 | 2 | 13 | 28 |
| 7 | 0.300000 | 17 | 68 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 16 | 817 | 6 | 3 | 13 | 28 |
| 7 | 0.300000 | 3 | 55750 | 6 | 6 | 25 | 28 |
| 7 | 0.300000 | 15 | 755 | 6 | 4 | 13 | 28 |
| 7 | 0.300000 | 6 | 31723 | 6 | 6 | 22 | 28 |
| 7 | 0.300000 | 13 | 944 | 6 | 4 | 15 | 28 |
| 7 | 0.300000 | 17 | 315 | 6 | 2 | 12 | 28 |
| 7 | 0.300000 | 5 | 3693 | 6 | 6 | 23 | 28 |
| 7 | 0.300000 | 15 | 25 | 6 | 2 | 11 | 28 |
| 7 | 0.300000 | 14 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.300000 | 16 | 260 | 6 | 4 | 11 | 28 |
| 7 | 0.300000 | 13 | 2332 | 6 | 3 | 15 | 28 |
| 7 | 0.300000 | 16 | 515 | 6 | 3 | 12 | 28 |
| 7 | 0.300000 | 13 | 2702 | 6 | 2 | 14 | 28 |
| 7 | 0.300000 | 15 | 718 | 6 | 3 | 14 | 28 |
| 7 | 0.300000 | 18 | 6 | 6 | 1 | 9 | 28 |
| 7 | 0.300000 | 16 | 109 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 16 | 15 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 19 | 14 | 6 | 1 | 9 | 28 |
| 7 | 0.300000 | 15 | 2751 | 6 | 4 | 14 | 28 |
| 7 | 0.300000 | 5 | 72122 | 6 | 6 | 23 | 28 |
| 7 | 0.300000 | 17 | 89 | 6 | 2 | 10 | 28 |
| 7 | 0.300000 | 17 | 99 | 6 | 2 | 11 | 28 |
| 7 | 0.300000 | 2 | 20577 | 6 | 6 | 25 | 28 |
| 7 | 0.300000 | 15 | 1315 | 6 | 2 | 13 | 28 |
| 7 | 0.300000 | 18 | 76 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 15 | 252 | 6 | 2 | 12 | 28 |
| 7 | 0.300000 | 17 | 112 | 6 | 3 | 10 | 28 |
| 7 | 0.300000 | 16 | 553 | 6 | 2 | 12 | 28 |
| 7 | 0.300000 | 17 | 108 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 17 | 225 | 6 | 1 | 11 | 28 |
| 7 | 0.300000 | 14 | 4367 | 6 | 4 | 15 | 28 |
| 7 | 0.300000 | 18 | 198 | 6 | 1 | 11 | 28 |
| 7 | 0.300000 | 16 | 284 | 6 | 2 | 11 | 28 |
| 7 | 0.300000 | 15 | 259 | 6 | 4 | 12 | 28 |
| 7 | 0.300000 | 17 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.300000 | 17 | 228 | 6 | 1 | 11 | 28 |
| 7 | 0.300000 | 5 | 31256 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.300000 | 17 | 69 | 6 | 1 | 10 | 28 |
| 7 | 0.300000 | 16 | 745 | 6 | 2 | 12 | 28 |
| 7 | 0.300000 | 18 | 76 | 6 | 2 | 10 | 28 |
| 7 | 0.300000 | 18 | 273 | 6 | 2 | 11 | 28 |
| 7 | 0.300000 | 13 | 958 | 6 | 4 | 15 | 28 |
| 7 | 0.300000 | 14 | 893 | 6 | 4 | 13 | 28 |
| 7 | 0.300000 | 16 | 494 | 6 | 2 | 12 | 28 |
| 7 | 0.350000 | 17 | 84 | 6 | 4 | 12 | 28 |
| 7 | 0.350000 | 15 | 593 | 6 | 2 | 13 | 28 |
| 7 | 0.350000 | 16 | 87 | 6 | 3 | 12 | 28 |
| 7 | 0.350000 | 6 | 19568 | 6 | 6 | 21 | 28 |
| 7 | 0.350000 | 12 | 69 | 6 | 2 | 13 | 28 |
| 7 | 0.350000 | 17 | 41 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 14 | 812 | 6 | 3 | 13 | 28 |
| 7 | 0.350000 | 4 | 29826 | 6 | 6 | 25 | 28 |
| 7 | 0.350000 | 15 | 770 | 6 | 4 | 13 | 28 |
| 7 | 0.350000 | 6 | 49439 | 6 | 6 | 22 | 28 |
| 7 | 0.350000 | 13 | 477 | 6 | 4 | 15 | 28 |
| 7 | 0.350000 | 15 | 291 | 6 | 2 | 12 | 28 |
| 7 | 0.350000 | 5 | 26267 | 6 | 6 | 23 | 28 |
| 7 | 0.350000 | 16 | 137 | 6 | 2 | 11 | 28 |
| 7 | 0.350000 | 14 | 709 | 6 | 3 | 13 | 28 |
| 7 | 0.350000 | 17 | 204 | 6 | 4 | 11 | 28 |
| 7 | 0.350000 | 13 | 2719 | 6 | 3 | 15 | 28 |
| 7 | 0.350000 | 15 | 430 | 6 | 3 | 12 | 28 |
| 7 | 0.350000 | 13 | 1441 | 6 | 2 | 14 | 28 |
| 7 | 0.350000 | 15 | 364 | 6 | 3 | 14 | 28 |
| 7 | 0.350000 | 19 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.350000 | 18 | 66 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 18 | 48 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 19 | 21 | 6 | 1 | 9 | 28 |
| 7 | 0.350000 | 14 | 2975 | 6 | 4 | 14 | 28 |
| 7 | 0.350000 | 5 | 72802 | 6 | 6 | 23 | 28 |
| 7 | 0.350000 | 18 | 95 | 6 | 2 | 10 | 28 |
| 7 | 0.350000 | 17 | 107 | 6 | 2 | 11 | 28 |
| 7 | 0.350000 | 4 | 20520 | 6 | 6 | 25 | 28 |
| 7 | 0.350000 | 15 | 922 | 6 | 2 | 13 | 28 |
| 7 | 0.350000 | 17 | 35 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 17 | 251 | 6 | 2 | 12 | 28 |
| 7 | 0.350000 | 17 | 104 | 6 | 3 | 10 | 28 |
| 7 | 0.350000 | 14 | 285 | 6 | 2 | 12 | 28 |
| 7 | 0.350000 | 17 | 37 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 15 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.350000 | 14 | 4511 | 6 | 4 | 15 | 28 |
| 7 | 0.350000 | 15 | 187 | 6 | 1 | 11 | 28 |
| 7 | 0.350000 | 16 | 118 | 6 | 2 | 11 | 28 |
| 7 | 0.350000 | 14 | 256 | 6 | 4 | 12 | 28 |
| 7 | 0.350000 | 18 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.350000 | 16 | 228 | 6 | 1 | 11 | 28 |
| 7 | 0.350000 | 7 | 51798 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.350000 | 18 | 112 | 6 | 1 | 10 | 28 |
| 7 | 0.350000 | 15 | 753 | 6 | 2 | 12 | 28 |
| 7 | 0.350000 | 18 | 71 | 6 | 2 | 10 | 28 |
| 7 | 0.350000 | 16 | 201 | 6 | 2 | 11 | 28 |
| 7 | 0.350000 | 13 | 3339 | 6 | 4 | 15 | 28 |
| 7 | 0.350000 | 16 | 1384 | 6 | 4 | 13 | 28 |
| 7 | 0.350000 | 14 | 329 | 6 | 2 | 12 | 28 |
| 7 | 0.400000 | 15 | 3 | 6 | 4 | 12 | 28 |
| 7 | 0.400000 | 15 | 593 | 6 | 2 | 13 | 28 |
| 7 | 0.400000 | 16 | 172 | 6 | 3 | 12 | 28 |
| 7 | 0.400000 | 8 | 19276 | 6 | 6 | 21 | 28 |
| 7 | 0.400000 | 14 | 277 | 6 | 2 | 13 | 28 |
| 7 | 0.400000 | 17 | 69 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 15 | 230 | 6 | 3 | 13 | 28 |
| 7 | 0.400000 | 3 | 42096 | 6 | 6 | 25 | 28 |
| 7 | 0.400000 | 15 | 816 | 6 | 4 | 13 | 28 |
| 7 | 0.400000 | 6 | 20324 | 6 | 6 | 22 | 28 |
| 7 | 0.400000 | 13 | 390 | 6 | 4 | 15 | 28 |
| 7 | 0.400000 | 15 | 447 | 6 | 2 | 12 | 28 |
| 7 | 0.400000 | 5 | 25885 | 6 | 6 | 23 | 28 |
| 7 | 0.400000 | 17 | 21 | 6 | 2 | 11 | 28 |
| 7 | 0.400000 | 15 | 653 | 6 | 3 | 13 | 28 |
| 7 | 0.400000 | 15 | 260 | 6 | 4 | 11 | 28 |
| 7 | 0.400000 | 13 | 2679 | 6 | 3 | 15 | 28 |
| 7 | 0.400000 | 16 | 514 | 6 | 3 | 12 | 28 |
| 7 | 0.400000 | 13 | 978 | 6 | 2 | 14 | 28 |
| 7 | 0.400000 | 15 | 284 | 6 | 3 | 14 | 28 |
| 7 | 0.400000 | 19 | 27 | 6 | 1 | 9 | 28 |
| 7 | 0.400000 | 17 | 100 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 16 | 20 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 18 | 15 | 6 | 1 | 9 | 28 |
| 7 | 0.400000 | 15 | 558 | 6 | 4 | 14 | 28 |
| 7 | 0.400000 | 5 | 72566 | 6 | 6 | 23 | 28 |
| 7 | 0.400000 | 18 | 56 | 6 | 2 | 10 | 28 |
| 7 | 0.400000 | 17 | 112 | 6 | 2 | 11 | 28 |
| 7 | 0.400000 | 4 | 46016 | 6 | 6 | 25 | 28 |
| 7 | 0.400000 | 15 | 849 | 6 | 2 | 13 | 28 |
| 7 | 0.400000 | 17 | 75 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 15 | 252 | 6 | 2 | 12 | 28 |
| 7 | 0.400000 | 17 | 77 | 6 | 3 | 10 | 28 |
| 7 | 0.400000 | 15 | 705 | 6 | 2 | 12 | 28 |
| 7 | 0.400000 | 18 | 30 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 15 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.400000 | 14 | 3781 | 6 | 4 | 15 | 28 |
| 7 | 0.400000 | 17 | 225 | 6 | 1 | 11 | 28 |
| 7 | 0.400000 | 16 | 132 | 6 | 2 | 11 | 28 |
| 7 | 0.400000 | 14 | 217 | 6 | 4 | 12 | 28 |
| 7 | 0.400000 | 18 | 18 | 6 | 1 | 9 | 28 |
| 7 | 0.400000 | 17 | 217 | 6 | 1 | 11 | 28 |
| 7 | 0.400000 | 7 | 50150 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.400000 | 18 | 54 | 6 | 1 | 10 | 28 |
| 7 | 0.400000 | 16 | 288 | 6 | 2 | 12 | 28 |
| 7 | 0.400000 | 18 | 76 | 6 | 2 | 10 | 28 |
| 7 | 0.400000 | 17 | 306 | 6 | 2 | 11 | 28 |
| 7 | 0.400000 | 13 | 2941 | 6 | 4 | 15 | 28 |
| 7 | 0.400000 | 14 | 903 | 6 | 4 | 13 | 28 |
| 7 | 0.400000 | 15 | 455 | 6 | 2 | 12 | 28 |
| 7 | 0.450000 | 17 | 123 | 6 | 4 | 12 | 28 |
| 7 | 0.450000 | 15 | 249 | 6 | 2 | 13 | 28 |
| 7 | 0.450000 | 15 | 87 | 6 | 3 | 12 | 28 |
| 7 | 0.450000 | 8 | 12521 | 6 | 6 | 21 | 28 |
| 7 | 0.450000 | 13 | 277 | 6 | 2 | 13 | 28 |
| 7 | 0.450000 | 18 | 41 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 14 | 812 | 6 | 3 | 13 | 28 |
| 7 | 0.450000 | 4 | 6868 | 6 | 6 | 25 | 28 |
| 7 | 0.450000 | 14 | 801 | 6 | 4 | 13 | 28 |
| 7 | 0.450000 | 6 | 20175 | 6 | 6 | 22 | 28 |
| 7 | 0.450000 | 13 | 2471 | 6 | 4 | 15 | 28 |
| 7 | 0.450000 | 17 | 285 | 6 | 2 | 12 | 28 |
| 7 | 0.450000 | 5 | 24445 | 6 | 6 | 23 | 28 |
| 7 | 0.450000 | 17 | 16 | 6 | 2 | 11 | 28 |
| 7 | 0.450000 | 15 | 653 | 6 | 3 | 13 | 28 |
| 7 | 0.450000 | 17 | 120 | 6 | 4 | 11 | 28 |
| 7 | 0.450000 | 12 | 2594 | 6 | 3 | 15 | 28 |
| 7 | 0.450000 | 15 | 220 | 6 | 3 | 12 | 28 |
| 7 | 0.450000 | 13 | 1428 | 6 | 2 | 14 | 28 |
| 7 | 0.450000 | 15 | 882 | 6 | 3 | 14 | 28 |
| 7 | 0.450000 | 18 | 6 | 6 | 1 | 9 | 28 |
| 7 | 0.450000 | 18 | 66 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 18 | 17 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 16 | 22 | 6 | 1 | 9 | 28 |
| 7 | 0.450000 | 15 | 2975 | 6 | 4 | 14 | 28 |
| 7 | 0.450000 | 5 | 72078 | 6 | 6 | 23 | 28 |
| 7 | 0.450000 | 18 | 112 | 6 | 2 | 10 | 28 |
| 7 | 0.450000 | 16 | 19 | 6 | 2 | 11 | 28 |
| 7 | 0.450000 | 4 | 37298 | 6 | 6 | 25 | 28 |
| 7 | 0.450000 | 15 | 460 | 6 | 2 | 13 | 28 |
| 7 | 0.450000 | 18 | 74 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 15 | 284 | 6 | 2 | 12 | 28 |
| 7 | 0.450000 | 17 | 109 | 6 | 3 | 10 | 28 |
| 7 | 0.450000 | 14 | 301 | 6 | 2 | 12 | 28 |
| 7 | 0.450000 | 17 | 9 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 15 | 201 | 6 | 1 | 11 | 28 |
| 7 | 0.450000 | 14 | 3633 | 6 | 4 | 15 | 28 |
| 7 | 0.450000 | 17 | 185 | 6 | 1 | 11 | 28 |
| 7 | 0.450000 | 17 | 119 | 6 | 2 | 11 | 28 |
| 7 | 0.450000 | 14 | 256 | 6 | 4 | 12 | 28 |
| 7 | 0.450000 | 18 | 28 | 6 | 1 | 9 | 28 |
| 7 | 0.450000 | 17 | 293 | 6 | 1 | 11 | 28 |
| 7 | 0.450000 | 7 | 51226 | 6 | 6 | 22 | 28 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 7 | 0.450000 | 16 | 20 | 6 | 1 | 10 | 28 |
| 7 | 0.450000 | 16 | 398 | 6 | 2 | 12 | 28 |
| 7 | 0.450000 | 18 | 91 | 6 | 2 | 10 | 28 |
| 7 | 0.450000 | 16 | 194 | 6 | 2 | 11 | 28 |
| 7 | 0.450000 | 13 | 1630 | 6 | 4 | 15 | 28 |
| 7 | 0.450000 | 16 | 1083 | 6 | 4 | 13 | 28 |
| 7 | 0.450000 | 15 | 423 | 6 | 2 | 12 | 28 |
| 8 | 0.000000 | 15 | 109175 | 7 | 7 | 21 | 36 |
| 8 | 0.000000 | 19 | 5747 | 7 | 4 | 17 | 36 |
| 8 | 0.000000 | 28 | 3995 | 7 | 2 | 16 | 36 |
| 8 | 0.000000 | 15 | 102311 | 7 | 4 | 22 | 36 |
| 8 | 0.000000 | 28 | 2484 | 7 | 4 | 16 | 36 |
| 8 | 0.000000 | 26 | 451 | 7 | 3 | 14 | 36 |
| 8 | 0.000000 | 24 | 98 | 7 | 2 | 12 | 36 |
| 8 | 0.000000 | 26 | 700 | 7 | 2 | 14 | 36 |
| 8 | 0.000000 | 27 | 1071 | 7 | 3 | 15 | 36 |
| 8 | 0.000000 | 14 | 243826 | 7 | 4 | 23 | 36 |
| 8 | 0.000000 | 20 | 6131 | 7 | 3 | 17 | 36 |
| 8 | 0.000000 | 25 | 176 | 7 | 4 | 13 | 36 |
| 8 | 0.000000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.000000 | 28 | 2509 | 7 | 3 | 16 | 36 |
| 8 | 0.000000 | 24 | 66 | 7 | 2 | 12 | 36 |
| 8 | 0.000000 | 25 | 65 | 7 | 1 | 11 | 36 |
| 8 | 0.000000 | 27 | 927 | 7 | 2 | 15 | 36 |
| 8 | 0.000000 | 14 | 88229 | 7 | 4 | 23 | 36 |
| 8 | 0.000000 | 25 | 209 | 7 | 2 | 12 | 36 |
| 8 | 0.000000 | 26 | 46 | 7 | 1 | 11 | 36 |
| 8 | 0.000000 | 27 | 1453 | 7 | 2 | 15 | 36 |
| 8 | 0.000000 | 19 | 9079 | 7 | 2 | 17 | 36 |
| 8 | 0.000000 | 28 | 3536 | 7 | 2 | 16 | 36 |
| 8 | 0.000000 | 29 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.000000 | 17 | 45074 | 7 | 7 | 19 | 36 |
| 8 | 0.000000 | 26 | 358 | 7 | 3 | 14 | 36 |
| 8 | 0.000000 | 15 | 136731 | 7 | 7 | 22 | 36 |
| 8 | 0.000000 | 15 | 76986 | 7 | 4 | 21 | 36 |
| 8 | 0.000000 | 26 | 34 | 7 | 1 | 10 | 36 |
| 8 | 0.000000 | 24 | 55 | 7 | 3 | 12 | 36 |
| 8 | 0.000000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.000000 | 25 | 176 | 7 | 3 | 13 | 36 |
| 8 | 0.000000 | 28 | 3654 | 7 | 2 | 16 | 36 |
| 8 | 0.000000 | 19 | 11228 | 7 | 3 | 17 | 36 |
| 8 | 0.000000 | 19 | 27267 | 7 | 5 | 18 | 36 |
| 8 | 0.000000 | 24 | 21 | 7 | 2 | 12 | 36 |
| 8 | 0.000000 | 27 | 1820 | 7 | 3 | 15 | 36 |
| 8 | 0.000000 | 26 | 27 | 7 | 1 | 10 | 36 |
| 8 | 0.000000 | 26 | 275 | 7 | 2 | 14 | 36 |
| 8 | 0.000000 | 28 | 2869 | 7 | 2 | 16 | 36 |
| 8 | 0.000000 | 16 | 95024 | 7 | 6 | 21 | 36 |
| 8 | 0.000000 | 26 | 670 | 7 | 2 | 14 | 36 |
| 8 | 0.000000 | 17 | 32901 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.000000 | 28 | 1975 | 7 | 3 | 16 | 36 |
| 8 | 0.000000 | 16 | 71146 | 7 | 6 | 21 | 36 |
| 8 | 0.000000 | 19 | 30205 | 7 | 2 | 18 | 36 |
| 8 | 0.000000 | 17 | 26244 | 7 | 7 | 19 | 36 |
| 8 | 0.000000 | 27 | 1268 | 7 | 3 | 15 | 36 |
| 8 | 0.000000 | 25 | 505 | 7 | 2 | 13 | 36 |
| 8 | 0.000000 | 17 | 26032 | 7 | 3 | 19 | 36 |
| 8 | 0.050000 | 15 | 109175 | 7 | 7 | 21 | 36 |
| 8 | 0.050000 | 19 | 2742 | 7 | 4 | 17 | 36 |
| 8 | 0.050000 | 28 | 3995 | 7 | 2 | 16 | 36 |
| 8 | 0.050000 | 15 | 102311 | 7 | 4 | 22 | 36 |
| 8 | 0.050000 | 29 | 5513 | 7 | 4 | 16 | 36 |
| 8 | 0.050000 | 26 | 451 | 7 | 3 | 14 | 36 |
| 8 | 0.050000 | 26 | 267 | 7 | 2 | 12 | 36 |
| 8 | 0.050000 | 26 | 700 | 7 | 2 | 14 | 36 |
| 8 | 0.050000 | 27 | 1071 | 7 | 3 | 15 | 36 |
| 8 | 0.050000 | 14 | 242819 | 7 | 4 | 23 | 36 |
| 8 | 0.050000 | 20 | 6131 | 7 | 3 | 17 | 36 |
| 8 | 0.050000 | 26 | 264 | 7 | 4 | 13 | 36 |
| 8 | 0.050000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.050000 | 30 | 7233 | 7 | 3 | 16 | 36 |
| 8 | 0.050000 | 24 | 66 | 7 | 2 | 12 | 36 |
| 8 | 0.050000 | 25 | 65 | 7 | 1 | 11 | 36 |
| 8 | 0.050000 | 27 | 927 | 7 | 2 | 15 | 36 |
| 8 | 0.050000 | 14 | 60677 | 7 | 4 | 23 | 36 |
| 8 | 0.050000 | 25 | 209 | 7 | 2 | 12 | 36 |
| 8 | 0.050000 | 26 | 46 | 7 | 1 | 11 | 36 |
| 8 | 0.050000 | 28 | 2474 | 7 | 2 | 15 | 36 |
| 8 | 0.050000 | 19 | 9079 | 7 | 2 | 17 | 36 |
| 8 | 0.050000 | 29 | 5826 | 7 | 2 | 16 | 36 |
| 8 | 0.050000 | 29 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.050000 | 17 | 45074 | 7 | 7 | 19 | 36 |
| 8 | 0.050000 | 26 | 358 | 7 | 3 | 14 | 36 |
| 8 | 0.050000 | 15 | 136675 | 7 | 7 | 22 | 36 |
| 8 | 0.050000 | 15 | 75106 | 7 | 4 | 21 | 36 |
| 8 | 0.050000 | 25 | 32 | 7 | 1 | 10 | 36 |
| 8 | 0.050000 | 24 | 55 | 7 | 3 | 12 | 36 |
| 8 | 0.050000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.050000 | 25 | 176 | 7 | 3 | 13 | 36 |
| 8 | 0.050000 | 29 | 4788 | 7 | 2 | 16 | 36 |
| 8 | 0.050000 | 19 | 11228 | 7 | 3 | 17 | 36 |
| 8 | 0.050000 | 19 | 27267 | 7 | 5 | 18 | 36 |
| 8 | 0.050000 | 24 | 21 | 7 | 2 | 12 | 36 |
| 8 | 0.050000 | 27 | 1820 | 7 | 3 | 15 | 36 |
| 8 | 0.050000 | 26 | 27 | 7 | 1 | 10 | 36 |
| 8 | 0.050000 | 26 | 275 | 7 | 2 | 14 | 36 |
| 8 | 0.050000 | 28 | 2869 | 7 | 2 | 16 | 36 |
| 8 | 0.050000 | 16 | 95024 | 7 | 6 | 21 | 36 |
| 8 | 0.050000 | 26 | 670 | 7 | 2 | 14 | 36 |
| 8 | 0.050000 | 17 | 32901 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.050000 | 28 | 1975 | 7 | 3 | 16 | 36 |
| 8 | 0.050000 | 16 | 71146 | 7 | 6 | 21 | 36 |
| 8 | 0.050000 | 19 | 30205 | 7 | 2 | 18 | 36 |
| 8 | 0.050000 | 17 | 26244 | 7 | 7 | 19 | 36 |
| 8 | 0.050000 | 27 | 1268 | 7 | 3 | 15 | 36 |
| 8 | 0.050000 | 25 | 505 | 7 | 2 | 13 | 36 |
| 8 | 0.050000 | 17 | 26032 | 7 | 3 | 19 | 36 |
| 8 | 0.100000 | 15 | 61362 | 7 | 7 | 21 | 36 |
| 8 | 0.100000 | 19 | 5747 | 7 | 4 | 17 | 36 |
| 8 | 0.100000 | 29 | 5011 | 7 | 2 | 16 | 36 |
| 8 | 0.100000 | 15 | 102311 | 7 | 4 | 22 | 36 |
| 8 | 0.100000 | 28 | 2484 | 7 | 4 | 16 | 36 |
| 8 | 0.100000 | 26 | 451 | 7 | 3 | 14 | 36 |
| 8 | 0.100000 | 25 | 144 | 7 | 2 | 12 | 36 |
| 8 | 0.100000 | 28 | 1357 | 7 | 2 | 14 | 36 |
| 8 | 0.100000 | 27 | 1071 | 7 | 3 | 15 | 36 |
| 8 | 0.100000 | 14 | 243826 | 7 | 4 | 23 | 36 |
| 8 | 0.100000 | 20 | 5751 | 7 | 3 | 17 | 36 |
| 8 | 0.100000 | 26 | 300 | 7 | 4 | 13 | 36 |
| 8 | 0.100000 | 27 | 1434 | 7 | 3 | 14 | 36 |
| 8 | 0.100000 | 29 | 4370 | 7 | 3 | 16 | 36 |
| 8 | 0.100000 | 24 | 66 | 7 | 2 | 12 | 36 |
| 8 | 0.100000 | 25 | 65 | 7 | 1 | 11 | 36 |
| 8 | 0.100000 | 27 | 927 | 7 | 2 | 15 | 36 |
| 8 | 0.100000 | 14 | 88229 | 7 | 4 | 23 | 36 |
| 8 | 0.100000 | 25 | 209 | 7 | 2 | 12 | 36 |
| 8 | 0.100000 | 26 | 59 | 7 | 1 | 11 | 36 |
| 8 | 0.100000 | 27 | 1542 | 7 | 2 | 15 | 36 |
| 8 | 0.100000 | 19 | 9079 | 7 | 2 | 17 | 36 |
| 8 | 0.100000 | 28 | 3536 | 7 | 2 | 16 | 36 |
| 8 | 0.100000 | 29 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.100000 | 17 | 45074 | 7 | 7 | 19 | 36 |
| 8 | 0.100000 | 27 | 722 | 7 | 3 | 14 | 36 |
| 8 | 0.100000 | 15 | 136731 | 7 | 7 | 22 | 36 |
| 8 | 0.100000 | 15 | 75106 | 7 | 4 | 21 | 36 |
| 8 | 0.100000 | 26 | 34 | 7 | 1 | 10 | 36 |
| 8 | 0.100000 | 24 | 55 | 7 | 3 | 12 | 36 |
| 8 | 0.100000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.100000 | 27 | 614 | 7 | 3 | 13 | 36 |
| 8 | 0.100000 | 28 | 3654 | 7 | 2 | 16 | 36 |
| 8 | 0.100000 | 19 | 4094 | 7 | 3 | 17 | 36 |
| 8 | 0.100000 | 19 | 27267 | 7 | 5 | 18 | 36 |
| 8 | 0.100000 | 24 | 21 | 7 | 2 | 12 | 36 |
| 8 | 0.100000 | 27 | 1820 | 7 | 3 | 15 | 36 |
| 8 | 0.100000 | 25 | 30 | 7 | 1 | 10 | 36 |
| 8 | 0.100000 | 26 | 275 | 7 | 2 | 14 | 36 |
| 8 | 0.100000 | 28 | 2869 | 7 | 2 | 16 | 36 |
| 8 | 0.100000 | 16 | 95024 | 7 | 6 | 21 | 36 |
| 8 | 0.100000 | 26 | 714 | 7 | 2 | 14 | 36 |
| 8 | 0.100000 | 17 | 32901 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.100000 | 29 | 3487 | 7 | 3 | 16 | 36 |
| 8 | 0.100000 | 16 | 71146 | 7 | 6 | 21 | 36 |
| 8 | 0.100000 | 19 | 30205 | 7 | 2 | 18 | 36 |
| 8 | 0.100000 | 17 | 26244 | 7 | 7 | 19 | 36 |
| 8 | 0.100000 | 29 | 3231 | 7 | 3 | 15 | 36 |
| 8 | 0.100000 | 27 | 866 | 7 | 2 | 13 | 36 |
| 8 | 0.100000 | 17 | 26032 | 7 | 3 | 19 | 36 |
| 8 | 0.150000 | 15 | 92487 | 7 | 7 | 21 | 36 |
| 8 | 0.150000 | 19 | 5740 | 7 | 4 | 17 | 36 |
| 8 | 0.150000 | 28 | 3995 | 7 | 2 | 16 | 36 |
| 8 | 0.150000 | 15 | 98522 | 7 | 4 | 22 | 36 |
| 8 | 0.150000 | 29 | 5536 | 7 | 4 | 16 | 36 |
| 8 | 0.150000 | 26 | 508 | 7 | 3 | 14 | 36 |
| 8 | 0.150000 | 25 | 144 | 7 | 2 | 12 | 36 |
| 8 | 0.150000 | 27 | 773 | 7 | 2 | 14 | 36 |
| 8 | 0.150000 | 30 | 3710 | 7 | 3 | 15 | 36 |
| 8 | 0.150000 | 14 | 243826 | 7 | 4 | 23 | 36 |
| 8 | 0.150000 | 20 | 5732 | 7 | 3 | 17 | 36 |
| 8 | 0.150000 | 25 | 143 | 7 | 4 | 13 | 36 |
| 8 | 0.150000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.150000 | 32 | 6599 | 7 | 3 | 16 | 36 |
| 8 | 0.150000 | 25 | 128 | 7 | 2 | 12 | 36 |
| 8 | 0.150000 | 24 | 78 | 7 | 1 | 11 | 36 |
| 8 | 0.150000 | 28 | 2606 | 7 | 2 | 15 | 36 |
| 8 | 0.150000 | 14 | 88087 | 7 | 4 | 23 | 36 |
| 8 | 0.150000 | 25 | 157 | 7 | 2 | 12 | 36 |
| 8 | 0.150000 | 26 | 46 | 7 | 1 | 11 | 36 |
| 8 | 0.150000 | 29 | 3162 | 7 | 2 | 15 | 36 |
| 8 | 0.150000 | 19 | 9069 | 7 | 2 | 17 | 36 |
| 8 | 0.150000 | 28 | 3536 | 7 | 2 | 16 | 36 |
| 8 | 0.150000 | 29 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.150000 | 17 | 45039 | 7 | 7 | 19 | 36 |
| 8 | 0.150000 | 26 | 358 | 7 | 3 | 14 | 36 |
| 8 | 0.150000 | 15 | 72561 | 7 | 7 | 22 | 36 |
| 8 | 0.150000 | 15 | 76671 | 7 | 4 | 21 | 36 |
| 8 | 0.150000 | 25 | 32 | 7 | 1 | 10 | 36 |
| 8 | 0.150000 | 24 | 65 | 7 | 3 | 12 | 36 |
| 8 | 0.150000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.150000 | 25 | 238 | 7 | 3 | 13 | 36 |
| 8 | 0.150000 | 28 | 3654 | 7 | 2 | 16 | 36 |
| 8 | 0.150000 | 19 | 4094 | 7 | 3 | 17 | 36 |
| 8 | 0.150000 | 19 | 23003 | 7 | 5 | 18 | 36 |
| 8 | 0.150000 | 25 | 125 | 7 | 2 | 12 | 36 |
| 8 | 0.150000 | 29 | 3338 | 7 | 3 | 15 | 36 |
| 8 | 0.150000 | 24 | 20 | 7 | 1 | 10 | 36 |
| 8 | 0.150000 | 26 | 275 | 7 | 2 | 14 | 36 |
| 8 | 0.150000 | 30 | 5533 | 7 | 2 | 16 | 36 |
| 8 | 0.150000 | 16 | 95024 | 7 | 6 | 21 | 36 |
| 8 | 0.150000 | 27 | 853 | 7 | 2 | 14 | 36 |
| 8 | 0.150000 | 17 | 32853 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.150000 | 30 | 6362 | 7 | 3 | 16 | 36 |
| 8 | 0.150000 | 16 | 69936 | 7 | 6 | 21 | 36 |
| 8 | 0.150000 | 19 | 29435 | 7 | 2 | 18 | 36 |
| 8 | 0.150000 | 17 | 14291 | 7 | 7 | 19 | 36 |
| 8 | 0.150000 | 27 | 1268 | 7 | 3 | 15 | 36 |
| 8 | 0.150000 | 26 | 801 | 7 | 2 | 13 | 36 |
| 8 | 0.150000 | 17 | 26032 | 7 | 3 | 19 | 36 |
| 8 | 0.200000 | 15 | 108979 | 7 | 7 | 21 | 36 |
| 8 | 0.200000 | 19 | 1232 | 7 | 4 | 17 | 36 |
| 8 | 0.200000 | 29 | 5011 | 7 | 2 | 16 | 36 |
| 8 | 0.200000 | 15 | 39538 | 7 | 4 | 22 | 36 |
| 8 | 0.200000 | 29 | 3840 | 7 | 4 | 16 | 36 |
| 8 | 0.200000 | 29 | 1985 | 7 | 3 | 14 | 36 |
| 8 | 0.200000 | 25 | 152 | 7 | 2 | 12 | 36 |
| 8 | 0.200000 | 26 | 720 | 7 | 2 | 14 | 36 |
| 8 | 0.200000 | 29 | 2780 | 7 | 3 | 15 | 36 |
| 8 | 0.200000 | 14 | 243814 | 7 | 4 | 23 | 36 |
| 8 | 0.200000 | 20 | 5179 | 7 | 3 | 17 | 36 |
| 8 | 0.200000 | 25 | 176 | 7 | 4 | 13 | 36 |
| 8 | 0.200000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.200000 | 28 | 2509 | 7 | 3 | 16 | 36 |
| 8 | 0.200000 | 25 | 143 | 7 | 2 | 12 | 36 |
| 8 | 0.200000 | 24 | 38 | 7 | 1 | 11 | 36 |
| 8 | 0.200000 | 29 | 3023 | 7 | 2 | 15 | 36 |
| 8 | 0.200000 | 12 | 60677 | 7 | 4 | 23 | 36 |
| 8 | 0.200000 | 25 | 147 | 7 | 2 | 12 | 36 |
| 8 | 0.200000 | 26 | 51 | 7 | 1 | 11 | 36 |
| 8 | 0.200000 | 29 | 3658 | 7 | 2 | 15 | 36 |
| 8 | 0.200000 | 19 | 8998 | 7 | 2 | 17 | 36 |
| 8 | 0.200000 | 28 | 3585 | 7 | 2 | 16 | 36 |
| 8 | 0.200000 | 26 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.200000 | 17 | 45038 | 7 | 7 | 19 | 36 |
| 8 | 0.200000 | 26 | 416 | 7 | 3 | 14 | 36 |
| 8 | 0.200000 | 15 | 72551 | 7 | 7 | 22 | 36 |
| 8 | 0.200000 | 15 | 76983 | 7 | 4 | 21 | 36 |
| 8 | 0.200000 | 25 | 28 | 7 | 1 | 10 | 36 |
| 8 | 0.200000 | 24 | 65 | 7 | 3 | 12 | 36 |
| 8 | 0.200000 | 15 | 44542 | 7 | 4 | 21 | 36 |
| 8 | 0.200000 | 25 | 238 | 7 | 3 | 13 | 36 |
| 8 | 0.200000 | 29 | 5707 | 7 | 2 | 16 | 36 |
| 8 | 0.200000 | 19 | 11172 | 7 | 3 | 17 | 36 |
| 8 | 0.200000 | 17 | 15891 | 7 | 5 | 18 | 36 |
| 8 | 0.200000 | 25 | 12 | 7 | 2 | 12 | 36 |
| 8 | 0.200000 | 27 | 1820 | 7 | 3 | 15 | 36 |
| 8 | 0.200000 | 26 | 12 | 7 | 1 | 10 | 36 |
| 8 | 0.200000 | 27 | 732 | 7 | 2 | 14 | 36 |
| 8 | 0.200000 | 29 | 5031 | 7 | 2 | 16 | 36 |
| 8 | 0.200000 | 16 | 84985 | 7 | 6 | 21 | 36 |
| 8 | 0.200000 | 27 | 1159 | 7 | 2 | 14 | 36 |
| 8 | 0.200000 | 17 | 13740 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.200000 | 30 | 4546 | 7 | 3 | 16 | 36 |
| 8 | 0.200000 | 16 | 71146 | 7 | 6 | 21 | 36 |
| 8 | 0.200000 | 19 | 29558 | 7 | 2 | 18 | 36 |
| 8 | 0.200000 | 17 | 26243 | 7 | 7 | 19 | 36 |
| 8 | 0.200000 | 27 | 1268 | 7 | 3 | 15 | 36 |
| 8 | 0.200000 | 26 | 867 | 7 | 2 | 13 | 36 |
| 8 | 0.200000 | 17 | 17620 | 7 | 3 | 19 | 36 |
| 8 | 0.250000 | 15 | 92291 | 7 | 7 | 21 | 36 |
| 8 | 0.250000 | 19 | 1827 | 7 | 4 | 17 | 36 |
| 8 | 0.250000 | 28 | 4765 | 7 | 2 | 16 | 36 |
| 8 | 0.250000 | 15 | 101841 | 7 | 4 | 22 | 36 |
| 8 | 0.250000 | 30 | 5113 | 7 | 4 | 16 | 36 |
| 8 | 0.250000 | 27 | 1117 | 7 | 3 | 14 | 36 |
| 8 | 0.250000 | 25 | 160 | 7 | 2 | 12 | 36 |
| 8 | 0.250000 | 28 | 1167 | 7 | 2 | 14 | 36 |
| 8 | 0.250000 | 29 | 2096 | 7 | 3 | 15 | 36 |
| 8 | 0.250000 | 14 | 158076 | 7 | 4 | 23 | 36 |
| 8 | 0.250000 | 20 | 5835 | 7 | 3 | 17 | 36 |
| 8 | 0.250000 | 27 | 377 | 7 | 4 | 13 | 36 |
| 8 | 0.250000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.250000 | 29 | 5086 | 7 | 3 | 16 | 36 |
| 8 | 0.250000 | 24 | 51 | 7 | 2 | 12 | 36 |
| 8 | 0.250000 | 24 | 37 | 7 | 1 | 11 | 36 |
| 8 | 0.250000 | 28 | 2502 | 7 | 2 | 15 | 36 |
| 8 | 0.250000 | 12 | 87689 | 7 | 4 | 23 | 36 |
| 8 | 0.250000 | 25 | 209 | 7 | 2 | 12 | 36 |
| 8 | 0.250000 | 26 | 46 | 7 | 1 | 11 | 36 |
| 8 | 0.250000 | 30 | 3137 | 7 | 2 | 15 | 36 |
| 8 | 0.250000 | 19 | 4673 | 7 | 2 | 17 | 36 |
| 8 | 0.250000 | 29 | 5189 | 7 | 2 | 16 | 36 |
| 8 | 0.250000 | 27 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.250000 | 17 | 45074 | 7 | 7 | 19 | 36 |
| 8 | 0.250000 | 28 | 1241 | 7 | 3 | 14 | 36 |
| 8 | 0.250000 | 13 | 65519 | 7 | 7 | 22 | 36 |
| 8 | 0.250000 | 15 | 12621 | 7 | 4 | 21 | 36 |
| 8 | 0.250000 | 25 | 34 | 7 | 1 | 10 | 36 |
| 8 | 0.250000 | 24 | 110 | 7 | 3 | 12 | 36 |
| 8 | 0.250000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.250000 | 27 | 569 | 7 | 3 | 13 | 36 |
| 8 | 0.250000 | 29 | 5228 | 7 | 2 | 16 | 36 |
| 8 | 0.250000 | 19 | 11228 | 7 | 3 | 17 | 36 |
| 8 | 0.250000 | 17 | 15888 | 7 | 5 | 18 | 36 |
| 8 | 0.250000 | 25 | 57 | 7 | 2 | 12 | 36 |
| 8 | 0.250000 | 29 | 3674 | 7 | 3 | 15 | 36 |
| 8 | 0.250000 | 26 | 27 | 7 | 1 | 10 | 36 |
| 8 | 0.250000 | 26 | 275 | 7 | 2 | 14 | 36 |
| 8 | 0.250000 | 29 | 5431 | 7 | 2 | 16 | 36 |
| 8 | 0.250000 | 16 | 86491 | 7 | 6 | 21 | 36 |
| 8 | 0.250000 | 30 | 1387 | 7 | 2 | 14 | 36 |
| 8 | 0.250000 | 17 | 31445 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.250000 | 29 | 3487 | 7 | 3 | 16 | 36 |
| 8 | 0.250000 | 16 | 67976 | 7 | 6 | 21 | 36 |
| 8 | 0.250000 | 19 | 29558 | 7 | 2 | 18 | 36 |
| 8 | 0.250000 | 17 | 26205 | 7 | 7 | 19 | 36 |
| 8 | 0.250000 | 28 | 2923 | 7 | 3 | 15 | 36 |
| 8 | 0.250000 | 25 | 565 | 7 | 2 | 13 | 36 |
| 8 | 0.250000 | 17 | 25259 | 7 | 3 | 19 | 36 |
| 8 | 0.300000 | 15 | 30902 | 7 | 7 | 21 | 36 |
| 8 | 0.300000 | 19 | 5738 | 7 | 4 | 17 | 36 |
| 8 | 0.300000 | 30 | 7787 | 7 | 2 | 16 | 36 |
| 8 | 0.300000 | 15 | 102311 | 7 | 4 | 22 | 36 |
| 8 | 0.300000 | 30 | 7375 | 7 | 4 | 16 | 36 |
| 8 | 0.300000 | 27 | 1096 | 7 | 3 | 14 | 36 |
| 8 | 0.300000 | 26 | 191 | 7 | 2 | 12 | 36 |
| 8 | 0.300000 | 28 | 1189 | 7 | 2 | 14 | 36 |
| 8 | 0.300000 | 29 | 3346 | 7 | 3 | 15 | 36 |
| 8 | 0.300000 | 14 | 158056 | 7 | 4 | 23 | 36 |
| 8 | 0.300000 | 20 | 5835 | 7 | 3 | 17 | 36 |
| 8 | 0.300000 | 27 | 391 | 7 | 4 | 13 | 36 |
| 8 | 0.300000 | 26 | 576 | 7 | 3 | 14 | 36 |
| 8 | 0.300000 | 32 | 9907 | 7 | 3 | 16 | 36 |
| 8 | 0.300000 | 24 | 45 | 7 | 2 | 12 | 36 |
| 8 | 0.300000 | 25 | 108 | 7 | 1 | 11 | 36 |
| 8 | 0.300000 | 28 | 2502 | 7 | 2 | 15 | 36 |
| 8 | 0.300000 | 12 | 87864 | 7 | 4 | 23 | 36 |
| 8 | 0.300000 | 26 | 238 | 7 | 2 | 12 | 36 |
| 8 | 0.300000 | 24 | 20 | 7 | 1 | 11 | 36 |
| 8 | 0.300000 | 28 | 2760 | 7 | 2 | 15 | 36 |
| 8 | 0.300000 | 19 | 4285 | 7 | 2 | 17 | 36 |
| 8 | 0.300000 | 30 | 7633 | 7 | 2 | 16 | 36 |
| 8 | 0.300000 | 24 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.300000 | 17 | 45074 | 7 | 7 | 19 | 36 |
| 8 | 0.300000 | 27 | 890 | 7 | 3 | 14 | 36 |
| 8 | 0.300000 | 15 | 136631 | 7 | 7 | 22 | 36 |
| 8 | 0.300000 | 15 | 76658 | 7 | 4 | 21 | 36 |
| 8 | 0.300000 | 26 | 17 | 7 | 1 | 10 | 36 |
| 8 | 0.300000 | 25 | 180 | 7 | 3 | 12 | 36 |
| 8 | 0.300000 | 15 | 52039 | 7 | 4 | 21 | 36 |
| 8 | 0.300000 | 26 | 373 | 7 | 3 | 13 | 36 |
| 8 | 0.300000 | 30 | 6911 | 7 | 2 | 16 | 36 |
| 8 | 0.300000 | 19 | 10002 | 7 | 3 | 17 | 36 |
| 8 | 0.300000 | 19 | 22933 | 7 | 5 | 18 | 36 |
| 8 | 0.300000 | 25 | 137 | 7 | 2 | 12 | 36 |
| 8 | 0.300000 | 28 | 2425 | 7 | 3 | 15 | 36 |
| 8 | 0.300000 | 25 | 32 | 7 | 1 | 10 | 36 |
| 8 | 0.300000 | 30 | 1758 | 7 | 2 | 14 | 36 |
| 8 | 0.300000 | 29 | 5431 | 7 | 2 | 16 | 36 |
| 8 | 0.300000 | 16 | 86775 | 7 | 6 | 21 | 36 |
| 8 | 0.300000 | 27 | 948 | 7 | 2 | 14 | 36 |
| 8 | 0.300000 | 17 | 23227 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.300000 | 29 | 3830 | 7 | 3 | 16 | 36 |
| 8 | 0.300000 | 14 | 58526 | 7 | 6 | 21 | 36 |
| 8 | 0.300000 | 19 | 30174 | 7 | 2 | 18 | 36 |
| 8 | 0.300000 | 17 | 19162 | 7 | 7 | 19 | 36 |
| 8 | 0.300000 | 28 | 2747 | 7 | 3 | 15 | 36 |
| 8 | 0.300000 | 25 | 550 | 7 | 2 | 13 | 36 |
| 8 | 0.300000 | 17 | 13484 | 7 | 3 | 19 | 36 |
| 8 | 0.350000 | 15 | 109043 | 7 | 7 | 21 | 36 |
| 8 | 0.350000 | 19 | 4835 | 7 | 4 | 17 | 36 |
| 8 | 0.350000 | 30 | 7848 | 7 | 2 | 16 | 36 |
| 8 | 0.350000 | 15 | 98624 | 7 | 4 | 22 | 36 |
| 8 | 0.350000 | 30 | 7375 | 7 | 4 | 16 | 36 |
| 8 | 0.350000 | 29 | 1880 | 7 | 3 | 14 | 36 |
| 8 | 0.350000 | 25 | 219 | 7 | 2 | 12 | 36 |
| 8 | 0.350000 | 27 | 1399 | 7 | 2 | 14 | 36 |
| 8 | 0.350000 | 29 | 3327 | 7 | 3 | 15 | 36 |
| 8 | 0.350000 | 12 | 45016 | 7 | 4 | 23 | 36 |
| 8 | 0.350000 | 18 | 5541 | 7 | 3 | 17 | 36 |
| 8 | 0.350000 | 27 | 468 | 7 | 4 | 13 | 36 |
| 8 | 0.350000 | 29 | 1709 | 7 | 3 | 14 | 36 |
| 8 | 0.350000 | 32 | 6254 | 7 | 3 | 16 | 36 |
| 8 | 0.350000 | 25 | 157 | 7 | 2 | 12 | 36 |
| 8 | 0.350000 | 25 | 108 | 7 | 1 | 11 | 36 |
| 8 | 0.350000 | 29 | 3088 | 7 | 2 | 15 | 36 |
| 8 | 0.350000 | 12 | 44148 | 7 | 4 | 23 | 36 |
| 8 | 0.350000 | 25 | 218 | 7 | 2 | 12 | 36 |
| 8 | 0.350000 | 24 | 28 | 7 | 1 | 11 | 36 |
| 8 | 0.350000 | 29 | 3078 | 7 | 2 | 15 | 36 |
| 8 | 0.350000 | 19 | 8964 | 7 | 2 | 17 | 36 |
| 8 | 0.350000 | 31 | 6456 | 7 | 2 | 16 | 36 |
| 8 | 0.350000 | 28 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.350000 | 17 | 26811 | 7 | 7 | 19 | 36 |
| 8 | 0.350000 | 26 | 358 | 7 | 3 | 14 | 36 |
| 8 | 0.350000 | 13 | 460 | 7 | 7 | 22 | 36 |
| 8 | 0.350000 | 14 | 75106 | 7 | 4 | 21 | 36 |
| 8 | 0.350000 | 24 | 20 | 7 | 1 | 10 | 36 |
| 8 | 0.350000 | 25 | 66 | 7 | 3 | 12 | 36 |
| 8 | 0.350000 | 15 | 43561 | 7 | 4 | 21 | 36 |
| 8 | 0.350000 | 26 | 580 | 7 | 3 | 13 | 36 |
| 8 | 0.350000 | 31 | 6989 | 7 | 2 | 16 | 36 |
| 8 | 0.350000 | 19 | 9946 | 7 | 3 | 17 | 36 |
| 8 | 0.350000 | 17 | 9543 | 7 | 5 | 18 | 36 |
| 8 | 0.350000 | 25 | 114 | 7 | 2 | 12 | 36 |
| 8 | 0.350000 | 28 | 1918 | 7 | 3 | 15 | 36 |
| 8 | 0.350000 | 25 | 2 | 7 | 1 | 10 | 36 |
| 8 | 0.350000 | 31 | 1500 | 7 | 2 | 14 | 36 |
| 8 | 0.350000 | 29 | 6615 | 7 | 2 | 16 | 36 |
| 8 | 0.350000 | 16 | 94992 | 7 | 6 | 21 | 36 |
| 8 | 0.350000 | 26 | 774 | 7 | 2 | 14 | 36 |
| 8 | 0.350000 | 17 | 5913 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.350000 | 32 | 9714 | 7 | 3 | 16 | 36 |
| 8 | 0.350000 | 16 | 52742 | 7 | 6 | 21 | 36 |
| 8 | 0.350000 | 17 | 2567 | 7 | 2 | 18 | 36 |
| 8 | 0.350000 | 17 | 26244 | 7 | 7 | 19 | 36 |
| 8 | 0.350000 | 30 | 3354 | 7 | 3 | 15 | 36 |
| 8 | 0.350000 | 27 | 637 | 7 | 2 | 13 | 36 |
| 8 | 0.350000 | 17 | 25296 | 7 | 3 | 19 | 36 |
| 8 | 0.400000 | 15 | 30968 | 7 | 7 | 21 | 36 |
| 8 | 0.400000 | 19 | 1060 | 7 | 4 | 17 | 36 |
| 8 | 0.400000 | 30 | 9565 | 7 | 2 | 16 | 36 |
| 8 | 0.400000 | 15 | 36296 | 7 | 4 | 22 | 36 |
| 8 | 0.400000 | 29 | 3840 | 7 | 4 | 16 | 36 |
| 8 | 0.400000 | 27 | 917 | 7 | 3 | 14 | 36 |
| 8 | 0.400000 | 24 | 106 | 7 | 2 | 12 | 36 |
| 8 | 0.400000 | 29 | 1377 | 7 | 2 | 14 | 36 |
| 8 | 0.400000 | 32 | 3741 | 7 | 3 | 15 | 36 |
| 8 | 0.400000 | 12 | 141195 | 7 | 4 | 23 | 36 |
| 8 | 0.400000 | 18 | 5224 | 7 | 3 | 17 | 36 |
| 8 | 0.400000 | 27 | 450 | 7 | 4 | 13 | 36 |
| 8 | 0.400000 | 30 | 1966 | 7 | 3 | 14 | 36 |
| 8 | 0.400000 | 31 | 8874 | 7 | 3 | 16 | 36 |
| 8 | 0.400000 | 28 | 207 | 7 | 2 | 12 | 36 |
| 8 | 0.400000 | 25 | 100 | 7 | 1 | 11 | 36 |
| 8 | 0.400000 | 28 | 1331 | 7 | 2 | 15 | 36 |
| 8 | 0.400000 | 14 | 72638 | 7 | 4 | 23 | 36 |
| 8 | 0.400000 | 26 | 293 | 7 | 2 | 12 | 36 |
| 8 | 0.400000 | 25 | 34 | 7 | 1 | 11 | 36 |
| 8 | 0.400000 | 30 | 3116 | 7 | 2 | 15 | 36 |
| 8 | 0.400000 | 19 | 4250 | 7 | 2 | 17 | 36 |
| 8 | 0.400000 | 32 | 6812 | 7 | 2 | 16 | 36 |
| 8 | 0.400000 | 28 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.400000 | 17 | 14425 | 7 | 7 | 19 | 36 |
| 8 | 0.400000 | 30 | 1583 | 7 | 3 | 14 | 36 |
| 8 | 0.400000 | 15 | 136693 | 7 | 7 | 22 | 36 |
| 8 | 0.400000 | 15 | 62167 | 7 | 4 | 21 | 36 |
| 8 | 0.400000 | 24 | 17 | 7 | 1 | 10 | 36 |
| 8 | 0.400000 | 24 | 16 | 7 | 3 | 12 | 36 |
| 8 | 0.400000 | 15 | 52040 | 7 | 4 | 21 | 36 |
| 8 | 0.400000 | 29 | 781 | 7 | 3 | 13 | 36 |
| 8 | 0.400000 | 18 | 2015 | 7 | 2 | 16 | 36 |
| 8 | 0.400000 | 19 | 9898 | 7 | 3 | 17 | 36 |
| 8 | 0.400000 | 19 | 20761 | 7 | 5 | 18 | 36 |
| 8 | 0.400000 | 29 | 355 | 7 | 2 | 12 | 36 |
| 8 | 0.400000 | 29 | 3338 | 7 | 3 | 15 | 36 |
| 8 | 0.400000 | 24 | 4 | 7 | 1 | 10 | 36 |
| 8 | 0.400000 | 28 | 1058 | 7 | 2 | 14 | 36 |
| 8 | 0.400000 | 30 | 6700 | 7 | 2 | 16 | 36 |
| 8 | 0.400000 | 16 | 82555 | 7 | 6 | 21 | 36 |
| 8 | 0.400000 | 27 | 777 | 7 | 2 | 14 | 36 |
| 8 | 0.400000 | 17 | 13264 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.400000 | 30 | 6345 | 7 | 3 | 16 | 36 |
| 8 | 0.400000 | 16 | 69856 | 7 | 6 | 21 | 36 |
| 8 | 0.400000 | 19 | 15067 | 7 | 2 | 18 | 36 |
| 8 | 0.400000 | 17 | 26200 | 7 | 7 | 19 | 36 |
| 8 | 0.400000 | 30 | 3195 | 7 | 3 | 15 | 36 |
| 8 | 0.400000 | 28 | 616 | 7 | 2 | 13 | 36 |
| 8 | 0.400000 | 17 | 21065 | 7 | 3 | 19 | 36 |
| 8 | 0.450000 | 15 | 95400 | 7 | 7 | 21 | 36 |
| 8 | 0.450000 | 19 | 2693 | 7 | 4 | 17 | 36 |
| 8 | 0.450000 | 32 | 10223 | 7 | 2 | 16 | 36 |
| 8 | 0.450000 | 13 | 101834 | 7 | 4 | 22 | 36 |
| 8 | 0.450000 | 29 | 4013 | 7 | 4 | 16 | 36 |
| 8 | 0.450000 | 31 | 2089 | 7 | 3 | 14 | 36 |
| 8 | 0.450000 | 25 | 311 | 7 | 2 | 12 | 36 |
| 8 | 0.450000 | 29 | 1397 | 7 | 2 | 14 | 36 |
| 8 | 0.450000 | 29 | 3055 | 7 | 3 | 15 | 36 |
| 8 | 0.450000 | 14 | 239850 | 7 | 4 | 23 | 36 |
| 8 | 0.450000 | 20 | 6087 | 7 | 3 | 17 | 36 |
| 8 | 0.450000 | 27 | 423 | 7 | 4 | 13 | 36 |
| 8 | 0.450000 | 27 | 1242 | 7 | 3 | 14 | 36 |
| 8 | 0.450000 | 31 | 6294 | 7 | 3 | 16 | 36 |
| 8 | 0.450000 | 24 | 83 | 7 | 2 | 12 | 36 |
| 8 | 0.450000 | 24 | 65 | 7 | 1 | 11 | 36 |
| 8 | 0.450000 | 30 | 3322 | 7 | 2 | 15 | 36 |
| 8 | 0.450000 | 12 | 87308 | 7 | 4 | 23 | 36 |
| 8 | 0.450000 | 28 | 314 | 7 | 2 | 12 | 36 |
| 8 | 0.450000 | 24 | 19 | 7 | 1 | 11 | 36 |
| 8 | 0.450000 | 29 | 3526 | 7 | 2 | 15 | 36 |
| 8 | 0.450000 | 19 | 5053 | 7 | 2 | 17 | 36 |
| 8 | 0.450000 | 31 | 7753 | 7 | 2 | 16 | 36 |
| 8 | 0.450000 | 24 | 1 | 7 | 0 | 8 | 36 |
| 8 | 0.450000 | 17 | 27514 | 7 | 7 | 19 | 36 |
| 8 | 0.450000 | 30 | 1646 | 7 | 3 | 14 | 36 |
| 8 | 0.450000 | 15 | 71748 | 7 | 7 | 22 | 36 |
| 8 | 0.450000 | 15 | 14254 | 7 | 4 | 21 | 36 |
| 8 | 0.450000 | 24 | 28 | 7 | 1 | 10 | 36 |
| 8 | 0.450000 | 26 | 311 | 7 | 3 | 12 | 36 |
| 8 | 0.450000 | 15 | 14204 | 7 | 4 | 21 | 36 |
| 8 | 0.450000 | 26 | 549 | 7 | 3 | 13 | 36 |
| 8 | 0.450000 | 33 | 7353 | 7 | 2 | 16 | 36 |
| 8 | 0.450000 | 19 | 1983 | 7 | 3 | 17 | 36 |
| 8 | 0.450000 | 19 | 22882 | 7 | 5 | 18 | 36 |
| 8 | 0.450000 | 27 | 339 | 7 | 2 | 12 | 36 |
| 8 | 0.450000 | 30 | 3710 | 7 | 3 | 15 | 36 |
| 8 | 0.450000 | 25 | 11 | 7 | 1 | 10 | 36 |
| 8 | 0.450000 | 27 | 607 | 7 | 2 | 14 | 36 |
| 8 | 0.450000 | 18 | 5124 | 7 | 2 | 16 | 36 |
| 8 | 0.450000 | 14 | 38355 | 7 | 6 | 21 | 36 |
| 8 | 0.450000 | 28 | 1562 | 7 | 2 | 14 | 36 |
| 8 | 0.450000 | 17 | 13717 | 7 | 3 | 19 | 36 |

*Continued on next page*

| pl | p | diff | n_path | E | largest_step | n_skipped | N |
|---|---|---|---|---|---|---|---|
| 8 | 0.450000 | 30 | 6345 | 7 | 3 | 16 | 36 |
| 8 | 0.450000 | 16 | 41332 | 7 | 6 | 21 | 36 |
| 8 | 0.450000 | 17 | 18796 | 7 | 2 | 18 | 36 |
| 8 | 0.450000 | 17 | 26239 | 7 | 7 | 19 | 36 |
| 8 | 0.450000 | 30 | 3567 | 7 | 3 | 15 | 36 |
| 8 | 0.450000 | 27 | 637 | 7 | 2 | 13 | 36 |
| 8 | 0.450000 | 17 | 23129 | 7 | 3 | 19 | 36 |