On The Average-Case Complexity of the Bottleneck Tower of Hanoi Problem*

Noam Solomon† Shay Solomon‡

Abstract

The Bottleneck Tower of Hanoi (BTH) problem, posed in 1981 by Wood, is a natural generalization of the classic Tower of Hanoi (TH) problem. There, a generalized placement rule allows a larger disk to be placed higher than a smaller one if their size difference is less than a given parameter $k \geq 1$; when $k = 1$ we arrive at the classic TH problem. The objective is to compute a shortest move-sequence transferring a legal (under the above rule) configuration of $n$ disks on three pegs to another legal configuration.

In SOFSEM’07, Dinitz and the second author established tight asymptotic bounds for the worst-case complexity of the BTH problem, for all $n$ and $k$. Moreover, they proved that the average-case complexity is asymptotically the same as the worst-case complexity, for all $n > 3k$ and $n \leq k$, and conjectured that the same phenomenon also occurs in the complementary range $k < n \leq 3k$.

In this paper we settle the conjecture of Dinitz and Solomon from SOFSEM’07 in the affirmative, and show that the average-case complexity of the BTH problem is asymptotically the same as the worst-case complexity, for all $n$ and $k$. To this end we provide a new proof that applies to all values of $n > k$. That is, our proof is not a patch over the previous proof of Dinitz and Solomon that is tailored only for the regime $k < n \leq 3k$, but is rather a stronger proof that is based on different principles and deeper ideas.

We also show that there are natural connections between the BTH problem, the problem of sorting with complete networks of stacks using a forklift [Albert and Atkinson 2002, König and Lübbecke 2008] and the well-studied pancake problem [Gates and Papadimitriou 1979].

1 Introduction

1.1 The (Classic) TH Problem and Configuration Graph. It is fascinating that the Tower of Hanoi (TH) problem still attracts the interest of mathematicians almost 130 years after its invention by the French number theorist Edouard Lucas (1842?1891). This stems from the rich inherent mathematical structure of the problem, which can be described in the following way.

We are given $n$ disks of sizes $1, 2, \ldots, n$ that are stacked on three vertical pegs, subject to the “divine rule”:

never to have a larger disk above a smaller one (on the same peg). A (legal) move is to pop the top-

most disk from one of the pegs and to push it to the top of one of the other two pegs, subject to the divine rule.

Consider the unweighted and undirected graph in which the nodes are all the (legal) configurations of the problem, and there is an edge between a pair of configurations if they are reachable via a single move; we refer to this graph as the TH configuration graph (or shortly, TH graph). The problem of interest is to find shortest paths in this graph, i.e., shortest move-
sequences transferring a given (initial) configuration to a given (final) configuration. In the most well-known case, both the initial and final configurations are perfect; in a perfect configuration all the disks are placed on a single peg in decreasing order of size, from disk $n$ at the bottom to disk 1 at the top. (The name of a disk is identified with its size.) It is easy to show that there is a unique shortest path in the graph between any pair of (distinct) perfect configurations, and the length of this path is $2^n - 1$. It is also long known that the length of the shortest path between any pair of arbitrary configurations is at most $2^n - 1$ [29, 11]; thus, the diameter of the TH graph is realized by a pair of distinct perfect configurations. Hinz [12] devised an algorithm for computing a shortest path between any pair of configurations; a more efficient algorithm was given later by Romik [24].

The connection between the TH problem and the Sierpiński gasket was first observed by Stewart [26]; in particular, the TH graph is isomorphic to the discrete Sierpiński gasket (see also [16, 24]). This connection was employed by Hinz and Schief [13] to conclude that the average distance on the Sierpiński gasket is $466/885$, or equivalently, the average distance between nodes in the TH graph is asymptotically $(1 + o(1))466/885 \cdot 2^n$; this result was also proved by Chan [4].

The TH graph was shown to be planar and (2-
-connected in [27]; it was proved to be Hamiltonian in [20], but it cannot be Eulerian, as there are always nodes with odd degree in the TH graph. For more detailed discussions on properties of the TH graph, see [23, 21], and the references therein.

1.2 The Bottleneck TH Problem and Configuration Graph. In 1981, D. Wood [28] suggested a natural generalization of the TH problem, character-
ized by the \( k \)-relaxed placement rule, \( k \geq 1 \): If disk \( j \) is placed higher than disk \( i \) on the same peg (not necessarily neighboring it), then their size difference \( j - i \) is less than \( k \). A move need no longer be subject to the (strict) divine rule, but rather to the \( k \)-relaxed placement rule. The objective remains unchanged, i.e., to find shortest paths in the induced configuration graph; note that when \( k = 1 \) we arrive at the classic problem. We refer to this problem as the Bottleneck Tower of Hanoi (BTH) problem (following Poole [22]), and denote it by \( BTH_n,k \); also, we refer to the induced configuration graph as the BTH configuration graph (or shortly, BTH graph), and denote it by \( G_{n,k}^{BTH} \). (See Figure 1 for an illustration.)

We remark that the number of all legal (under the \( k \)-relaxed placement rule) configurations increases with \( k \). For example, for \( k = 1 \), the perfect configuration of the \( n \) disks is the only legal configuration where all \( n \) disks lie on a specific peg, whereas for \( k \geq n \), all \( n! \) permutations of the \( n \) disks on that peg are legal.

Poole [22] suggested a natural algorithm for computing a shortest path in the BTH graph between any pair of perfect configurations, for all values of \( n \) and \( k \), but the question whether this algorithm is optimal was left open. Beneditikis, Berend, and Safro [2] proved Poole’s algorithm to be optimal for the first non-trivial case \( k = 2 \) only. Optimality of Poole’s algorithm in the general case was proved independently by Dinitz and Solomon [6, 8] and by Chen et al. [5]. It was proved in [7] that there is more than one shortest path in the BTH graph between any pair of perfect configurations, for all \( k \geq 2 \); also, a complete characterization of the set of all such shortest paths was given therein, complemented with a closed formula, depending on \( n \) and \( k \), for the cardinality of this set.

Denote the diameter of the BTH graph \( G_{n,k}^{BTH} \) by \( Diam(n,k) \). Tight asymptotic bounds for \( Diam(n,k) \) were established in [7].

**Theorem 1.1.** (Theorem 2 in [7])

\[
Diam(n,k) = \begin{cases} 
  \Theta(n \cdot \log n) & \text{if } n \leq k, \\
  \Theta(k \cdot \log k + (n-k)^2) & \text{if } k < n \leq 2k, \\
  \Theta(k^2 \cdot 2^{\frac{k}{2}}) & \text{if } n > 2k.
\end{cases}
\]

We remark that the upper bound proof of Theorem 1.1 is constructive. That is, given an arbitrary pair \( C, C' \) of configurations in \( G_{n,k}^{BTH} \), a move-sequence of length that is bounded above by \( Diam(n,k) \) (as given in the upper bound of Theorem 1.1) transferring \( C \) to \( C' \) is provided. Refer to [25] for further detail; see Theorem 3.2.1 therein, and the corresponding proof.

We denote by \( Avg(n,k) \) the average distance between nodes in \( G_{n,k}^{BTH} \). Notice that \( Avg(n,k) \leq Diam(n,k) \), for all values of \( n \) and \( k \).

The following theorem from [7] asserts that \( Avg(n,k) \) and \( Diam(n,k) \) are asymptotically the same, for all values of \( n \leq k \) and \( n > 3k \).

**Theorem 1.2.** (Theorem 3 in [7]) For all values of \( n \leq k \) and \( n > 3k \), \( Avg(n,k) = \Theta(Diam(n,k)) \).

It was conjectured in [7] that \( Avg(n,k) \) and \( Diam(n,k) \) are asymptotically the same also in the complementary range \( k < n \leq 3k \). (See Conjecture 1 in [7].)

We remark that the regime \( n \leq k \) and \( n > 3k \) that was handled in [7] is easier to analyze than the complementary range \( k < n \leq 3k \) that was left open therein. Indeed, in the extreme case \( n \leq k \), the \( k \)-relaxed placement rule poses no restrictions; it is easy to analyze this somewhat degenerate case, where pegs behave as stacks. It is also easy to analyze the other extreme case \( k = 1 \) (more generally, \( k \ll n \)), where there is only one legal way (more generally, only few legal ways) to place any disk set on a peg. The number of legal configurations increases with \( k \); intuitively—the more freedom we have in placing disks, the more difficult the problem gets. Summarizing, the informal conclusion of this intuitive discussion is that the most difficult case of the BTH problem is when \( k \) approaches \( n \) from below—specifically, the regime \( k < n \leq 3k \).

In this paper we settle the conjecture of [7] in the affirmative, and conclude:

**Theorem 1.3.** For all values of \( n \) and \( k \), \( Avg(n,k) = \Theta(Diam(n,k)) \).

To prove Theorem 1.3 we provide a new proof that applies to all values of \( n > k \). That is, our proof is not a patch over the previous proof of [7] that is tailored only for the regime \( k < n \leq 3k \), but is rather a stronger proof that is based on different principles and deeper ideas. See Sections 1.3 and 3 for further details.

### 1.3 Stack Sorting, Fork Stacks, and the Pancake Problem

Consider an arbitrary configuration in the BTH graph \( G_{n,k}^{BTH} \), and some path between this configuration and a perfect one on some fixed peg. The move-sequence corresponding to this path can be viewed as a “sorting sequence” that rearranges the \( n \) disks in the “correct order” on the fixed peg. If we can find a sorting sequence for any configuration, then we have at hand a “sorting algorithm”. Note that by applying the sorting algorithm twice, we can transfer any configuration in the BTH graph to any other (not necessarily perfect) configuration.

In the case \( n \leq k \), the pegs behave as stacks. The problem of stack sorting was introduced by Knuth [17], and has received much attention in the literature (see [3, 15, 18, 9], and the references therein). Variants
of this problem include imposing restrictions on the legal moves (e.g., a disk that has been popped from a stack may never be pushed back to it again) and considering more than three stacks; in fact, another variant of stack sorting that has been studied in [19, 18] is a natural generalization of the BTH problem, where the placement of disks on a single stack is subject to constraints that are modeled by a conflict graph.

In the case $n > k$ things become more complicated. **Some notation.** Consider a move $M$ of some disk from peg $X$ to peg $Y$; the pegs $X$ and $Y$ are the source and destination pegs of $M$, respectively, whereas the third peg $Z \neq X, Y$ is called the spare peg of $M$. A move $M$ in a move-sequence $S$ is called switched (with respect to $S$) if (i) it is not the first move of $S$, and (ii) the spare peg of $M$ is different from the spare peg of the preceding move in $S$. Consider a move-sequence $S$ of the disk set $\{1, 2, \ldots, n\}$, and restrict attention to the $k'$ largest disks $\{n - k' + 1, n - k' + 2, \ldots, n\}$, for some (small) parameter $k'$. Notice that any disk in this set cannot be placed higher than any disk in $\{1, 2, \ldots, n - k' + 1\}$ on the same peg. Thus, it is easy to see that if $S$ contains $\ell$ switched moves of the $k'$ largest disks, then at least $\ell$ packet-moves of the disk set $\{1, 2, \ldots, n - k' + 1\}$ are required; a packet-move of a disk set $D$ is a move-sequence transferring the entire disk set $D$ from one peg to another. Finally, the $k'$-switched distance between a pair $C, C'$ of configurations in the BTH graph $G_{n,k}^{\text{BTH}}$ is defined as the minimum number of switched moves of the $k'$ largest disks in any move-sequence transferring $C$ to $C'$. Dinitz and Solomon [7] made a critical use of this notion of switched distance to obtain both upper and lower bounds for $\text{Diam}(n, k)$ and $\text{Avg}(n, k)$. In particular, they proved (see Lemma 2 in [7]) that the average $2k$-switched distance, taken over all nodes of $G_{n,k}^{\text{BTH}}$, is $O(k)$. This lemma of [7] constitutes the heart of the lower bound analysis of [7]. Specifically, it implies that $\Omega(k)$ packet-moves of the disk set $\{1, 2, \ldots, n - 3k + 1\}$ are required on the average; consequently, a tight asymptotic lower bound on $\text{Avg}(n, k)$ was derived in [7], for the range $n > 3k$ only. To prove a tight lower bound on $\text{Avg}(n, k)$ in the entire range of $n > k$, we strengthen Lemma 2 of [7] significantly by showing (see Theorem 3.1 in Section 3) that for any parameter $k' = 1, 2, \ldots, O(k)$ (rather than just for the particular case $k' = 2k$), the average $k'$-switched distance, taken over all nodes of $G_{n,k}^{\text{BTH}}$, is $\Omega(k')$. We stress that this improvement from the particular case $k' = 2k$ to any parameter $k' = O(k)$ requires a far deeper understanding of the BTH problem—it constitutes the core of the problem, and should not be viewed as just a small technicality.

There is a close connection between the notion of switched distance and the notion of fork stack [1]. Rather than moving a single element from (the top of) one stack to (the top of) another, a fork stack is equipped with a forklift that can be used for moving multiple elements from one peg to another in a single step. It is easy to see that in the case $n \leq k$, the $n$-switched distance between a pair of configurations in $G_{n,k}^{\text{BTH}}$ is (essentially) equal to the minimum number of steps needed to get between these configurations using three fork stacks. In addition, the notions of switched distance and fork stack are closely related to the well-studied pancake problem, where the objective is to sort permutations by prefix reversal (see [10, 14], and the references therein). We remark that an upper bound on the number $f(n)$ of prefix reversals required to sort a permutation of $n$ elements provides the same (up to a constant factor) upper bound on the $n$-switched distance between a pair of gathered configurations in $G_{n,k}^{\text{BTH}}$, $n \leq k$; a configuration is called gathered if all $n$ disks lie on the same peg.

### 1.4 Structure of the Paper

In Section 2 we present the notation that is used throughout the paper (in addition to the notation that was already provided in Section 1.3). Section 3 is devoted to the proof of Theorem 1.3. We start (Section 3.1) with proving a statement that is central in the proof of Theorem 1.3, and then employ this statement (Section 3.2) to complete the proof of the theorem. Finally, in Section 4 we
Let \( \text{Theorem 3.1.} \)

For any disk set \( D \), any configuration \( C \) of \( D \), and any \( D' \subseteq D \), the restriction \( C|_{D'} \) is \( C \) with all disks not in \( D' \) removed. Note that if \( C \) is legal (under the \( k \)-relaxed placement rule), then its restriction \( C|_{D'} \) is legal as well.

For a pair of positive integers \( w, z \), such that \( w \leq z \), we denote the sets \( \{w, w+1, \ldots, z\} \) and \( \{1, 2, \ldots, z\} \) by \([w, z]\) and \([z]\), respectively. The entire disk set \([n]\) is divided into \( \lceil \frac{n}{3} \rceil \) blocks \( B_i = B_i(n) \): \( B_1 = [(n-k+1),n] \), \( B_2 = [(n-2k+1),(n-k)] \), \ldots, \( B_{\lceil \frac{n}{3} \rceil} = [1, (n-(\lceil \frac{n}{3} \rceil-1)\cdot k)] \). Note that the set of disks in any block is allowed to be placed on the same peg in an arbitrary order. For any \( n \geq 1 \), let \( \text{Small}(n) \) denote the set \([n] \setminus B_1(n) \).

We will use the following result of [6,8] in the sequel.

**Theorem 2.1.** (Theorem 3.2 in [8]) Let \( n = sk + r \), where \( 0 \leq r < k \), and define \( b_{n,k} = (k+r) \cdot 2^s \cdot k \). Under the rules of \( BTH \), the length of any packet-move of \([n]\) is at least \( b_{n,k} \).

**Remark:** It is easy to see that \( b_{n,k} = \Theta(k \cdot 2^s) \), for all \( n \geq k \).

### 3 Proof of Theorem 1.3

In this section we prove that \( \text{Avg}(n,k) = \Omega(Diam(n,k)) \), for all \( n > k \). Since \( \text{Avg}(n,k) \leq \text{Diam}(n,k) \), for all values of \( n \) and \( k \), Theorem 1.3 would follow.

The following statement is central in our proof.

**Theorem 3.1.** Let \( k' \leq k \). For at least half of all pairs of legal configurations of \([n]\), the minimum number of switched moves of disks in \([n-k'+1,n]\) required to get from one configuration to another is at least \( \frac{k'}{2} - 2 \).

In Section 3.1 we prove Theorem 3.1. We then employ Theorem 3.1 in Section 3.2 to complete the proof of Theorem 1.3.

#### 3.1 Proof of Theorem 3.1

To prove Theorem 3.1, we show that for any legal configuration \( C_{init} \), the minimum number of switched moves of disks in \([n-k'+1,n]\) required to get from \( C_{init} \) at to at least half the legal configurations of \([n]\) is \( \frac{k'}{2} - 2 \).

**3.1.1 Dividing the disks of each peg into triads.** Let \( B^k_1(n) \), \( B^k_2(n) \) denote the \( k' \)-largest disks of \( B_1(n), B_2(n) \), respectively. In case \( k' = k \), these sets coincide with \( B_1(n), B_2(n) \), respectively.

For \( a, b, c \in B^k_1(n) \), we call the ordered triple \((a, b, c)\) a triad.

Consider some configuration \( C' \) of \([n]\), and let \( C'|_{B^k_1(n)} \) be its restriction to \( B^k_1(n) \). For each peg \( X \), let us divide the set of disks in \( C'|_{B^k_1(n)} \) on \( X \) into triads and a residue of size at most 9, according to their placement at \( C'|_{B^k_1(n)} \), as follows. Each triad consists of three consecutive disks of \( B^k_1(n) \) placed on the same peg, \( X \), except for the top-most triad at each peg (which may contain fewer than three disks) that we disregard; the reason we disregard the top-most triad at each peg will become clear soon. Let \( l_{C'} \) denote the number of such triads (disregarding the top-most triad at each peg); these triads are referred to as the triads of \( C' \).

Since we disregard at most one triad at each peg, we have

\[
l_{C'} \geq \frac{k'}{3} - 3.
\]

A disk is called switched w.r.t. a move-sequence \( S \) if it participates in at least one switched move. (See Section 1.3 for the definition of switched move; recall that we restrict the attention to the \( k' \)-largest disks, i.e., to the disks in \( B^k_1(n) \).) We say that a triad is switched w.r.t. a move-sequence \( S \) from \( C_{init} \) to \( C' \), if at least one disk in that triad is switched w.r.t. \( S \). A triad is called cheap, if the disks in it at \( C_{init} \) are consecutive on some peg, and either preserve their order at \( C' \), or reverse it; otherwise, it is called expensive.

The following claim was proved in [7].

**Claim 3.1.** Any expensive triad in \( C' \) is switched w.r.t. any move-sequence from \( C_{init} \) to \( C' \).

**3.1.2 Reducing the problem.** The problem is reduced (using Claim 3.1) to showing that for at least half the legal configurations, \( C' \), there are at least \( \frac{k'}{2} - 2 \) expensive triads in \( C' \) (w.r.t. \( C_{init} \)).

Having partitioned the disks of \( C'|_{B^k_1(n)} \) into triads, usually there will be other disks in \( C' \) placed in the “spaces” between the disks of these triads. For each triad \((a, b, c)\) (not necessarily corresponding to a specific configuration), the collection of three spaces: above \( a \) (but below the higher triads on that peg), between \( a \) and \( b \), and between \( b \) and \( c \), is referred to as the envelope of the triad \((a, b, c)\).

Since we disregarded the top-most triad at each peg, the disks in the envelope of any triad cannot belong to \([n-k-k'] = [n] \setminus (B_1(n) \cup B^k_2(n)) \); indeed, otherwise the \( k \)-relaxed placement rule would be violated.

A triad \((a, b, c)\) together with disks from \( B^k_2(n) \) placed in its envelope, is called a completed triad; note that we disregard disks from \( B_1(n) \setminus B^k_1(n) \) in the envelope of triad \((a, b, c)\), or in other words, according to this definition a completed triad consists of three disks from \( B^k_1(n) \) and some (possibly zero) disks from \( B^k_2(n) \).
Lemma 3.1. (A counting lemma) Given \(a_1, \ldots, a_p, Y\) as above, with \(a_1 < a_2 < \ldots < a_p\), let \(\ell_i\) be the number of legal ways to place disk \(a_i\) in one of the spaces between the disks of \(Y\). Then \(N(a_1, \ldots, a_p, Y) = \prod_{i=1}^p (\ell_i + i - 1)\).

Proof. The proof is via induction on \(p\). The basis of induction \(p = 1\) is clear. For the induction step, we assume the claim holds for \(p - 1\) and prove it for \(p \leq k'\). Assume first that \(a_1, \ldots, a_{p-1}\) were legally inserted. To count the number of legal ways to insert an additional disk \(a_p\), we first notice that there are \(\ell_p\) legal placements of \(a_p\) when ignoring the disks \(a_1, \ldots, a_{p-1}\). Since \(a_p\) is larger than \(a_1, \ldots, a_{p-1}\) and smaller than all disks in \(B_k^p(n)\), each one of \(a_i, i = 1, \ldots, p - 1\), that were inserted gives an additional legal placement for \(a_p\). This gives \(p - 1\) additional legal placements for \(a_p\), giving \(\ell_p + p - 1\) legal placements for \(a_p\) once the \(a_1, \ldots, a_{p-1}\) were legally placed. By the induction hypothesis, the number of legal configurations of \(\{a_1, \ldots, a_{p-1}\}\) placed in its associated envelope is \(\prod_{i=1}^{p-1} (\ell_i + i - 1)\). Multiplying it by \(\ell_p + p - 1\) completes the proof of Lemma 3.1.

Claim 3.2. We have \(l_{v'} \geq \frac{k'}{12} - 3\).

Proof. For each non-sparse triad, there are at least 4 disks from \(B_k^3(n)\) placed in its envelope. As \(|B_k^3(n)| = k'\), there can be at most \(\frac{k'}{4}\) non-sparse triads. By Equation (3.1) there are at least \(\frac{k'}{4} - 3\) triads, and subtracting from it at most \(\frac{k'}{4}\) non-sparse triads completes the proof of the claim.

Remark: Claim 3.2 shows that there are sufficiently many sparse triads, hence we can restrict the attention to sparse triads in the sequel. This observation is very useful, since sparse triads are much more convenient for analysis purposes than arbitrary triads.

For any \(a, b, c\) in \(B_k^3(n)\), let \(f(a, b, c)\), the face of \((a, b, c)\), denote the union of the triads \((a, b, c)\) and \((c, b, a)\). The 6 distinct permutations of \(\{a, b, c\}\) are divided into three faces \(f(a, b, c), f(a, c, b), f(b, a, c)\).

With the above notation, when \(Y = (a, b, c)\) is a triad, and \(0 \leq p \leq 3\), let

\[
N_{f(a,b,c)}^{a_1, \ldots, a_p} := N(a_1, \ldots, a_p, (a, b, c)) + N(a_1, \ldots, a_p, (c, b, a))
\]

denote the number of legal ways to complete the triads of the face \(f(a, b, c)\) w.r.t. the disks \(a_1, \ldots, a_p\) being placed in its associated envelope. Throughout this section, we assume \(0 \leq p \leq 3\).

Proposition 3.1. (Bottleneck triangle inequality) With the above notation

\[
N_{f(a,b,c)}^{a_1, \ldots, a_p} \leq N_{f(a,c,b)}^{a_1, \ldots, a_p} + N_{f(b,a,c)}^{a_1, \ldots, a_p}.
\]

Proof. Assume without loss of generality that \(a_1 < \ldots < a_p\). We next employ Lemma 3.1 to provide a formula for the number of legal ways to complete a triad \((a, b, c)\) w.r.t. \(p\) disks \(a_1, \ldots, a_p\) placed in its envelope; this formula depends on the monotone non-decreasing sequence \((s_1, \ldots, s_p)\), \(0 \leq s_i \leq 3\), where \(s_i\) denotes below how many of \(\{a, b, c\}\) can \(a_i\) be legally placed. In what follows let \(A < B < C\) be the ordering of \((a, b, c)\).

Lemma 3.2. With the above notation, the number of legal completions of the triad \((a, b, c)\) with respect to \(a_1, \ldots, a_p\) placed in its associated envelope, namely \(N(a_1, \ldots, a_p, (a, b, c))\), is

\[
\begin{align*}
\Pi_{i=1}^p (\min(s_i, 2) + i), & \quad \text{for } (a, b, c) = (A, B, C), \\
\Pi_{i=1}^p (2s_i + i), & \quad \text{for } (a, b, c) = (C, B, A), \\
\Pi_{i=1}^p (\min(s_i, 1) + s_{i+1} + i), & \quad \text{for } (a, b, c) = (C, A, B) = (A, C, B), \\
\Pi_{i=1}^p (\max(1, s_i + i - 1)), & \quad \text{for } (a, b, c) = (B, C, A), \\
\Pi_{i=1}^p (2s_i), & \quad \text{for } (a, b, c) = (C, A, B) \text{ (same as } (a, b, c) = (C, B, A)), \\
\Pi_{i=1}^p (2s_i + 2 + i), & \quad \text{for } (a, b, c) = (B, A, C).
\end{align*}
\]

(Here for a predicate \(\text{pred}\), let \(\chi_{\text{pred}}\) denote the integer value \(1\) if \(\text{pred}\) is true, and \(0\) otherwise.)

Proof. Assume \(a_1 < \ldots < a_p\). Let \(\ell_i\) denote the number of legal ways to place disk \(a_i\) in the envelope associated to the triad \((a, b, c)\). The space above \(a\) is always legal. The number \(\ell_i\) of legal ways to place each \(a_i\) is determined by the value of \(s_i\) and the ordering of \((a, b, c)\). (For instance, if \((a, b, c) = (A, B, C)\) then \(\ell_i = \min(s_i, 2) + 1\).) By Lemma 3.1, letting \(Y = \{A, B, C\}\), the number of ways to complete the
trial \((A, B, C)\) w.r.t. \(a_1, \ldots, a_p\) placed in its envelope is
\(N(a_1, \ldots, a_p, Y) = \prod_{i=1}^{r}(\ell_i + i - 1)\). Representing each
\(\ell_i\) as a function of \(s_i\) and then plugging the resulting
terms in \(\prod_{i=1}^{r}(\ell_i + i - 1)\), for each ordering of \((a, b, c)\),
completes the proof of Lemma 3.2.

Next, we continue the proof of Proposition 3.1.

We show that the face \(f(A, B, C)\) satisfies the
Bottleneck triangle inequality, i.e., that
\[
N^{a_1, \ldots, a_p}_{f(A,B,C)} \leq N^{a_1, \ldots, a_p}_{f(A,C,B)} + N^{a_1, \ldots, a_p}_{f(B,A,C)}
\]
As \(N(a_1, \ldots, a_p, (C, B, A)) = N(a_1, \ldots, a_p, (C, A, B))\),

it reduces to proving that \(N(a_1, \ldots, a_p, (A, B, C)) < \)
\(N(a_1, \ldots, a_p, (A, C, B)) + N(a_1, \ldots, a_p, (B, A, C)) + \)
\(N(a_1, \ldots, a_p, (B, C, A)).\)
We divide into cases according to the \((s_1, \ldots, s_p)\)-sequence.
If all \(s_i \leq 1\) then
\[N(a_1, \ldots, a_p, (A, B, C)) = N(a_1, \ldots, a_p, (A, C, B)).\]
If \(s_p \geq 2\), whereas \(s_i \leq 1\) for \(i < p\), Lemma 3.2 implies that
\[
N(a_1, \ldots, a_p, (A, B, C)) = \left( \prod_{i=1}^{p-1}(s_i + i) \right)(2 + p)
\leq \left( \prod_{i=1}^{p-1}(s_i + i) \right)(1 + p)
+ \left( \prod_{i=1}^{p-1}i \right)(2 + p)
+ \left( \prod_{i=1}^{p-1}i \right)(1 + p)
\leq N(a_1, \ldots, a_p, (A, C, B))
+ N(a_1, \ldots, a_p, (B, A, C))
+ N(a_1, \ldots, a_p, (B, C, A)).
\]
The remaining case is \(p = 3, s_1 \leq 1, s_2, s_3 \geq 2\), where
\[
N(a_1, \ldots, a_p, (A, B, C)) = (s_1 + 1) \cdot 4 
< (s_1 + 1) \cdot 3 \cdot 4 + 4 \cdot 5
+ 3 \cdot 4
\leq N(a_1, \ldots, a_p, (A, C, B))
+ N(a_1, \ldots, a_p, (B, A, C))
+ N(a_1, \ldots, a_p, (B, C, A)).
\]
This proves that the face \(f(A, B, C)\) satisfies the Bottleneck
triangle inequality, as required. It can be verified in the same way
that the faces \(f(A, C, B)\) and \(f(B, A, C)\) also satisfy the Bottleneck triangle inequality.
Proposition 3.1 follows.

3.1.5 Replacing cheap completed triads with
expensive ones is now easy. Given \(a, b, c \in B^k_{\gamma}(n)\),
and \(a_1, \ldots, a_p \in B^k_{\gamma}(n), 0 \leq p \leq 3\), let \(T^{a_1, \ldots, a_p}_{a,b,c}\)
(respectively, \(\check{C}^{a_1, \ldots, a_p}_{a,b,c} ; \check{C}^{a_1, \ldots, a_p}_{a,b,c} \))
de note the collection of possible (resp., cheap; expensive) completions of triads
on the disks \(a, b, c\) w.r.t. the disks \(a_1, \ldots, a_p\) placed in
its envelope. (Notice that all orderings of \(a, b, c\) are allowed here.)

When it causes no confusion, we will omit the subscript \(a, b, c\)
and the superscript \(a_1, \ldots, a_p\), or in other words, we will write \(T, C\) and \(E\) as a shortcut
for \(T^{a_1, \ldots, a_p}_{a,b,c}\) and \(\check{C}^{a_1, \ldots, a_p}_{a,b,c}\), respectively.

The Bottleneck triangle inequality (Proposition 3.1) implies that
(independently of any specific configuration)

**Corollary 3.1.** With the above notation, there exists
a bijective map
\[
f^{a_1, \ldots, a_p}_{a,b,c} \cdot T^{a_1, \ldots, a_p}_{a,b,c} \rightarrow T^{a_1, \ldots, a_p}_{a,b,c}
\]
such that \(f(C) \cap C = \emptyset, f^2 = id, f|_{T \setminus (C \cup f(C))} = id\). In
other words, \(f\) sends a cheap completed triad \(t\) to an
expensive one, it sends \(f(t)\) back to \(t\), and the other
trials of \(T\) remain untouched by \(f\).

Next, we observe that for an illegal configuration \(C\) of \([n]\), there exist
two disks \(a\) and \(b\) on the same peg, such that \(a\) is located somewhere above \(b\)
and \(a \geq b + k\).
Such an incidence is referred to as a clash of \(a\) and \(b\),
and we say that \(a\) clashes with \(b\).

Given a legal configuration \(C'\) of \([n]\), a (legal) completed triad \(t \in T^{a_1, \ldots, a_p}_{a,b,c}\) belonging to \(C'\),
and some other (legal) completed triad \(t' \in T^{a_1, \ldots, a_p}_{a,b,c}\), let \((C' : t \leftrightarrow t')\) denote the configuration of \([n]\) obtained
by changing the order of the \(3 + p\) elements belonging
to \(t\) according to their order in \(t'\) (both \(t\) and \(t'\) have
the same elements), and leaving the other disks of \(C'\)
untouched.
Lemma 3.3. With the above notation, \((C' : t \leftrightarrow t')\) is a legal configuration of \([n]\).

Proof. If \((C' : t \leftrightarrow t')\) is an illegal configuration of \([n]\), then there is a clash involving some disk of \(t\). First, in each peg of \(C'\), the disks in \([n - k - k']\) are placed above the highest (disregarded) triad on this peg. This implies that these disks cannot clash with any disk of the permuted triad. Second, disks belonging to \(B_1(n) \setminus B^n_1(n)\) cannot clash with disks from \(B^n_2(n)\) and \(B^n_3(n)\), as the absolute value of the difference would not exceed \(k\). Third, there can be no clashes between the disks of \(t\) and disks of other completed triads on the same peg, as \(C'\) is legal. Finally, there can be no inner clashes between the disks of \(t\), as \(t'\) is a legal completed triad. Thus, there can be no clash at all, completing the proof of Lemma 3.3.

With the above notation, let
\[
\tilde{f}(C', t) := (C' : t \leftrightarrow f(t)).
\]

Lemma 3.3 then implies

Corollary 3.2. \(\tilde{f}(C', t)\) is a legal configuration of \([n]\).

Applying the map \(\tilde{f}\) iteratively we obtain

Lemma 3.4. Given a legal configuration \(C_{init}\), there exists a bijection \(F_{init}\) (depending on \(C_{init}\)), on the set of legal configuration, such that for a legal configuration \(C'\) of \([n]\),

1. The configuration \(F_{init}(C')\) is legal.

2. The number of expensive triads of \(C'\) together with the number of expensive triads of \(F_{init}(C')\) is at least as the number of sparse triads of \(C'\).

Proof. 1. For the configuration \(C'\), let \(\{t_i\}, i = 1, \ldots, l^s_{C'}\), denote the collection of completed sparse triads in it, listed from bottom to top on the three pegs, one after the other. We define \(F_{init}\) by applying \(\tilde{f}\) iteratively on \(C'\) and the completed triads, or more formally,

\[
F^0_{init}(C') := C',
\]

\[
F^i_{init}(C') := \tilde{f}(F^{i-1}_{init}(C'), t_i), i = 1, \ldots, l^s_{C'}.
\]

Let

\[
F_{init}(C') := F^{l^s_{C'}}_{init}(C').
\]

By Corollary 3.2, standard induction implies that each \(F_{init}(C')\) is a legal configuration of \([n]\), and therefore so is \(F_{init}(C')\).

Since \(f^2 = id\), it is easy to verify that \(F^2_{init} = id\), implying in particular that \(F_{init}\) is a bijection.

2. Corollary 3.1 implies that each cheap (completed) triad is mapped under \(F_{init}\) to an expensive one. Hence this part of the lemma is clear, and the proof of Lemma 3.4 is complete.

By Claim 3.2, there are at least \(\frac{k'}{24} - 3\) sparse triads in each legal configuration \(C'\) of \([n]\). Consequently, Lemma 3.4 implies that either \(C'\) or \(F_{init}(C')\) must have at least \(\frac{k'}{24} - 2\) expensive triads. This concludes the proof of Theorem 3.1.

3.2 Completing the Proof of Theorem 1.3 Having proved Theorem 3.1, we now turn to prove the desired lower bounds on \(Avg(n, k)\) for all values of \(n > k\), thus completing the proof of Theorem 1.3.

Notice that it suffices to restrict attention to the range \(k < n \leq 3k\), which was left open in [7]. Nevertheless, since our lower bounds on \(Avg(n, k)\) in the entire range \(n > k\) are derived more or less in the same way—as a rather simple corollary of Theorem 3.1, we provide here a lower bound proof that applies to the entire range \(n > k\). Our lower bound proof for \(Avg(n, k)\) is stronger than the original proof of [7] which applies only to \(n > 3k\), and can therefore replace it.

Lemma 3.5. For all \(n > k\), \(Avg(n, k) = \Omega(k \cdot \log k)\).

Proof. The statement is trivial if \(k = \Omega(1)\). We henceforth assume that \(k\) is super-constant.

Let \(C\) be an arbitrary legal configuration of \([n]\), i.e., an arbitrary node in the BTH graph \(G_{n, k}^{BTH}\). Consider a breadth-first search (BFS) tree \(T_C\) of \(G_{n, k}^{BTH} = (V, E)\) rooted at \(C\). To prove the lemma, it suffices to show that the average distance between \(C\) and all other nodes in \(T_C\) is \(\Omega(k \cdot \log k)\).

It is easy to see that the number of all legal (under the \(k\)-relaxed placement rule) configurations of \([n]\) is at least \(\Omega(k!\ldots)\), yielding \(|V| = \Omega(k!)\). Observe that the maximum degree of \(G_{n, k}^{BTH}\) is at most \(6\), hence the maximum degree of \(T_C\) is at most \(6\) as well. It follows that at most \(\frac{1}{6}(6^i - 1)\) nodes are at distance at most \(i\) in \(T_C\) from the root \(C\), for each index \(i \geq 0\). Substituting \(i = \lceil \log_6 |V| \rceil - 1\), we conclude that at least \(\frac{1}{6} \cdot |V|\) nodes are at distance at least \(\lceil \log_6 |V| \rceil = \Omega(k \cdot \log k)\) in \(T_C\) from \(C\). The lemma follows.

Lemma 3.6. For all \(k < n \leq 3k\), \(Avg(n, k) = \Omega((n - k)^2)\).

Proof. Define \(k' = \frac{n - k}{2}\), and note that \(k' \leq k\). The statement is trivial if \(k' = O(1)\). We henceforth assume that \(k'\) is super-constant.
By Theorem 3.1, for at least half of all pairs of legal configurations of \([n]\), at least \(k + 1\) switched moves of disks in \([n - k' + 1, n]\) are required to get from one configuration to another. Observe that for any integer \(\ell \geq 1\), any move-sequence of \([n]\) that contains \(\ell\) switched moves of disks in \([n - k' + 1, n]\) requires at least \(\ell\) packet-moves of the disk set \(\text{Small}(n - k' + 1) = [n - k' + 1, n]\). By Theorem 2.1, each packet-move of \([n - k' + 1, n]\) requires at least \(\frac{n - k}{2}\). It follows that at least \((k' - 2)\cdot \frac{n - k}{2} + 1\) moves are required to get between at least half of all pairs of legal configurations of \([n]\). Hence \(\text{Avg}(n, k) = \Omega((n - k)^2)\), and we are done.

The proof of the following lemma is very similar to the proof of Lemma 3.6. We provide it for completeness.

**Lemma 3.7.** For all \(n > 3k\), \(\text{Avg}(n, k) = \Omega(k^2 \cdot 2^k)\).

**Proof.** For \(k = O(1)\), the lemma can be easily proved using Theorem 2.1. We henceforth consider the more interesting case, namely, when \(k\) is super-constant.

By Theorem 3.1 (in the particular case \(k' = k\)), for at least half of all pairs of legal configurations of \([n]\), at least \(\frac{k^2}{2} - 2\) switched moves of disks in \([n - k + 1, n]\) are required to get from one configuration to another. For any integer \(\ell \geq 1\), any move-sequence of \([n]\) that contains \(\ell\) switched moves of disks in \([n - k + 1, n]\) requires at least \(\ell\) packet-moves of the disk set \(\text{Small}(n - k + 1) = [n - k + 1, n]\). By Theorem 2.1, each packet-move of \([n - 2k + 1, n]\) requires at least \(\frac{n - 2k + 1}{2}\) moves. It follows that at least \((\frac{n}{2} - 2)\cdot \frac{n - 2k + 1}{2} + 1\) moves are required to get between at least half of all pairs of legal configurations of \([n]\). It follows that \(\text{Avg}(n, k) = \Omega(k^2 \cdot 2^k)\). \(\square\)

Lemmas 3.5, 3.6 and 3.7 imply that

\[
\text{Avg}(n, k) = \begin{cases} 
\Omega(k \cdot \log k + (n - k)^2) & \text{if } k < n \leq 2k, \\
\Omega(k^2 \cdot 2^k) & \text{if } n > 2k.
\end{cases}
\]

By Theorem 1.1, we conclude that \(\text{Avg}(n, k) = \Omega(\text{Diam}(n, k))\), for all \(n > k\). Theorem 1.3 now follows.

**4 Future Work**

We conclude the paper by outlining some directions for future work.

1. Disregarding constant factors, Theorems 1.1 and 1.3 provide tight bounds for \(\text{Diam}(n, k)\) and \(\text{Avg}(n, k)\), for all values of \(n\) and \(k\). A challenging open problem is to determine the precise constant factors that are hidden within the \(\Theta\)-notation of these bounds. As mentioned in Section 1, for the special case \(k = 1\), the precise constant factors are known.

2. Another intriguing question is to find a pair of configurations that realize the diameter \(\text{Diam}(n, k)\), for general values of \(n\) and \(k\). For the special case \(k = 1\), it is known that the diameter is realized by a pair of perfect configurations (on different pegs). However, it is not difficult to see that this phenomenon does not generalize for larger values of \(k\). We believe that this question is of particular interest in the case \(n \leq k\), where the pegs behave as stacks.

3. Another natural question is to obtain closed formulae, depending on \(n\) and \(k\), for the following two quantities: (i) The number of all legal configurations of \([n]\) on the three pegs, i.e., the number of nodes in the BTH graph \(G_{n,k}^{\text{BTH}}\). (ii) The number of all (legal) gathered configurations of \([n]\). For very small values of \(k\) (say, \(k \leq 4\)), we can obtain closed formulae for these two quantities by solving recurrence relations; for example, for \(k = 2\), the number of all gathered configurations of \([n]\) is given by the \(((n + 1)\)th Fibonacci number \(F_{n+1}\). However, obtaining closed formulae for general values of \(n\) and \(k\) should be significantly more difficult.

4. Theorem 1.1 provides tight asymptotic bounds for \(\text{Diam}(n, k)\). As was mentioned in Section 1.2, the upper bound proof of this theorem is constructive. That is, given an arbitrary pair \(C, C'\) of configurations in \(G_{n,k}^{\text{BTH}}\), a move-sequence of length that is bounded above by \(\text{Diam}(n, k)\) (as given in the upper bound of Theorem 1.1) transferring \(C\) to \(C'\) is provided. It would be interesting to devise an algorithm for computing a shortest path between any pair of configurations in \(G_{n,k}^{\text{BTH}}\). In Section 1.1 we mentioned that for the special case \(k = 1\), such an algorithm was given in [12, 24]. If computing a shortest path in the general case \(k \geq 2\) is too difficult, one can settle for a path whose length is greater than the shortest one by a “sufficiently small” factor; this question in the particular case of \(n \leq k\) coincides with an open question on stacks that was raised in [18].

5. Finally, we believe that investigating additional properties of the BTH graph \(G_{n,k}^{\text{BTH}}\) is a promising direction for future work.
References