VARIATIONS ON THE FOUR-POST TOWER OF HANOI PUZZLE

THESIS

Presented to the Graduate Council of
Southwest Texas State University
in Partial Fulfillment of
the Requirements

For the Degree of

Master of SCIENCE

By

Steven Greenstein, B.S.

San Marcos, Texas
August 2003

Dedicated to my father and the memory of my mother

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Jian Shen, for his confidence in my ability when he first awarded me the research assistantship from which this project evolved. It was because of his encouragement that I have had such an incredible and meaningful experience. I, too, wish to thank him for his generous help and valuable suggestions. I would also like each of the other members of my thesis committee for their valuable comments and suggestions, all of which have made this a better thesis: I wish to thank Dr. Eugene Curtin for his frequent assistance with programming in *Mathematica*. Without his assistance this project would not have been as fruitful nor would I have learned so much. And thank you, also, to Dr. Xingde Jia for similarly helpful assistance with LaTeX. I am so grateful for every one of his comments, each of which has made this thesis clearer and stronger.

I wish to thank my partner, Shana, for encouragement and understanding as I relentlessly pursued the kind of educational experience I always wanted.
This manuscript was submitted on May 6, 2003.

TABLE OF CONTENTS

LIST OF FIGURES

Page

ABSTRACT

VARIATIONS OF THE FOUR-POST TOWER OF HANOI PUZZLE

by

STEVEN GREENSTEIN, B.S.

Southwest Texas State University

June 2003

SUPERVISING PROFESSOR: JIAN SHEN

The Tower of Hanoi puzzle consists of three posts and a set of n, typically eight, pierced disks of differing diameters that can be stacked on the posts. The tower is formed initially by stacking the disks onto one post in decreasing order of size from bottom to top. The challenge is to transport the tower to another post by moving the disks one at a time from one post to another, subject to the rule that no disk can ever be placed on top of a smaller disk. The disks may move from one post to any other without restriction. The objective of the puzzle is to complete the task of transporting the tower from one post to another in the minimum number of moves. This problem has been solved for a puzzle with three posts.

This thesis considers two variations of the classic Tower of Hanoi puzzle on four posts:

1. **Four-in-a-row puzzle**, in which the allowable moves are, in either direction, between posts A and B, posts B and C, posts C and D; and

2. **Four-post cyclic puzzle** in which disks may only be moved clockwise along

a directed cycle, from post A to post B, from B to C, and from D to A.

We develop an algorithm for the four-in-a-row puzzle where the number of moves is of order $O\left(4.72881^{\sqrt{n}}\right)$ and an algorithm for the four-post cyclic puzzle where the number of moves is of order $O\left(2.34^n\right)$. Each of these results improves the best known results proposed by Stockmeyer in 1994 [8] and by Scorer, Grundy, and Smith in 1944 [7] where the minimum number of moves achieved by each of these algorithms is $O(3^n)$.

CHAPTER 1

INTRODUCTION

The Tower of Hanoi puzzle (sometimes referred to as the Tower of Brahma or the End of the World Puzzle) was invented in 1883 by Édouard Lucas. He was inspired by a legend that tells of a Hindu temple where the pyramid puzzle might have been used for the mental discipline of young priests. Legend says that at the beginning of time the priests in the temple were given a stack of 64 gold disks, each one a little smaller than the one beneath it. Their assignment was to transfer the 64 disks from one of the three posts to another, with one important proviso: a large disk could never be placed on top of a smaller one. The priests worked very efficiently, day and night. When they finished their work, the myth said, the temple would crumble into dust and the world would vanish [4]. Today the puzzle is best known in computer programming for demonstrating the power of recursion in problem solving [2].

The Tower consists of three posts and a set of n, typically eight, pierced disks of differing diameters that can be stacked on the posts. The tower is formed initially by stacking the disks onto one post in decreasing order of size from bottom to top. The challenge is to transport the tower to another post by moving the disks one at a time from one post to another, subject to the rule that no disk can ever be placed on top of a smaller disk.

Many variations of this puzzle have been proposed in which the set of allowable moves has been extended or restricted, the number of posts has changed, or some other aspect has been varied [8]. In this thesis, we study two versions of the puzzle that use four posts, rather than three: the cyclic puzzle and the four-in-a-row puzzle. [See Figure 1.]

We will begin by considering the classic 3-post puzzle mentioned and recall disks may move from any one post to any other on their way from post A to post C.

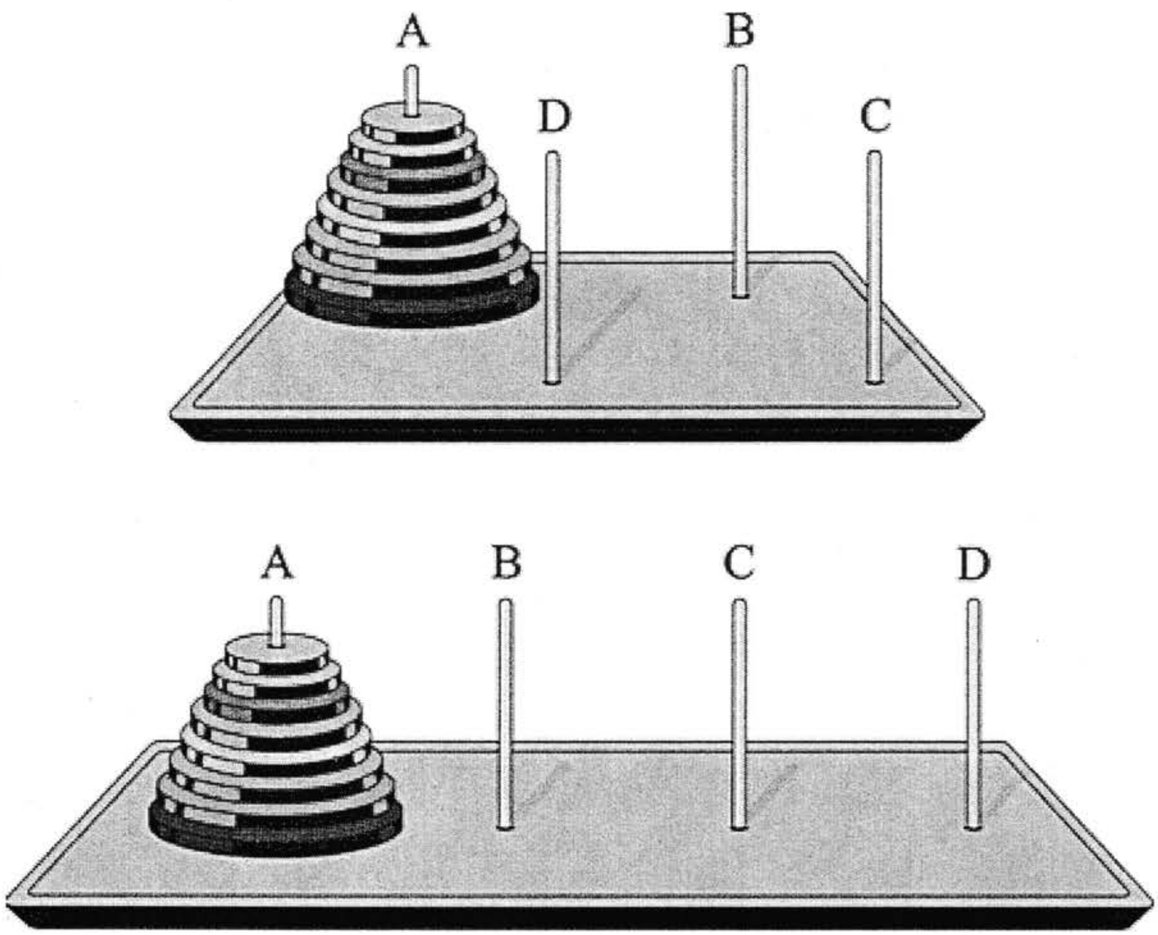

Figure 1: **The 4-post cyclic and 4-in-a-row puzzles**

With respect to this classic puzzle, it is well known that $2^n - 1$ moves are necessary and sufficient to transport n disks two steps from post A, the starting point, to post C. In order to introduce the reader to the type of algorithms that appear in this thesis and to typical methods used to solve them, we demonstrate how this result was achieved.

The algorithm used to transport the tower of height n from post A to post C where disks may move one at a time and from any one post to any other on their way from post A to post C subject only to the typical constraint that no disk can ever be placed on top of a smaller disk is as follows:

Algorithm 1 : **Classic 3-post puzzle**

1. *Recursively transport the $n-1$ smallest disks from post A to post B (using*

post C as an intermediary post);

2. *Move the largest disk from post A to post C in one move;*

3. *Recursively transport the $n-1$ smallest disks from post B to post C (using post A as an intermediary post).*

In various algorithms throughout this thesis, we will use $f(n)$ to denote the number of moves required according to the current algorithm. The number of moves required by each step above is as follows:

- Step 1: $f(n-1)$,
- Step 2: 1,
- Step 3: $f(n-1)$.

So $f(n) = 2f(n-1) + 1$ and $f(0) = 0$.
With respect to this relation, we see by induction:

$$\begin{aligned} f(n) &= 2f(n-1) + 1 \\ &= 2^2 f(n-2) + 2 + 1 \\ &= 2^3 f(n-3) + 2^2 + 2 + 1 \\ &= \cdots \\ &= 2^n f(0) + 2^{n-1} + 2^{n-2} + \cdots + 2^0 \\ &= 2^n - 1 \end{aligned}$$

Thus we have obtained the recurrence relation:

$$f(n) = 2f(n-1) + 1 \text{ with } f(1) = 1$$

whose solution is

$$f(n) = 2^n - 1.$$

Once we complete consideration of the three-post puzzle, we will move on to variations of the four-post puzzle, originally proposed by Henry Dudeney in his book, *The Canterbury Puzzles* [1]. In order to entertain the pilgrims on their way to Canterbury, the Reve posed the problem of conveying a stack of cheeses of varying sizes from the first of four stools to the last, moving the cheeses one at a time from any stool to any other, without ever putting any cheese on top of a smaller one. Dudeney states without proof that the number of moves needed to convey a stack of size 8, 10, or 21 is 33, 49, or 321, respectively [8]. We will consider such a four-post puzzle, one that we call the four-in-a-row puzzle, with the additional constraint that cheeses (or disks) may only move to the adjacent post(s).

We will complete our study of variations of the Hanoi puzzle by considering the variation of the four-post puzzle, one we call the four-post cyclic puzzle, where posts are arranged in a circle and disks may move counterclockwise, one post at a time.

In general, a puzzle with m posts and n disks is referred to as the multi-peg Tower of Hanoi puzzle. Most researchers believe that an algorithm proposed in 1941 by Frame and Stewart in *American Mathematical Monthly* [3] (generally referred to as Frame's Algorithm) is optimal [6]. Several references exist in which authors have tried to solve the multi-peg problem only to rediscover "Frame's Conjecture" and so today that conjecture is to be "presumed optimal" [4,8]. An outline of that algorithm is presented in Chapter 3 upon discussion of that puzzle.

CHAPTER 2

THREE-IN-A-ROW PUZZLE

A variation of the 3-post puzzle proposed in 1944 by Scorer, Grundy, and Smith [7] restricts the movements of disks, in either direction, between posts A and B and post B and C. Moves are not allowed between posts A and C. We will refer to this algorithm as the "two-step 3-in-a-row algorithm."

We illustrate Scorer, Grundy, and Smith's proposed algorithm and verify their result by generating a recurrence relation and solving it by induction. Again, this is the technique we will use throughout this thesis to develop closed formulas that describe the number of moves required by each algorithm.

Algorithm 2 : Scorer, Grundy, and Smith's Two-step, 3-in-a-row algorithm: (Also see Figure 2.)

First, we remark on the base case as follows: if $n = 1$, this algorithm requires two steps. When describing subsequent algorithms, we will assume an analogous base case (for $n = 1$ disk). Then we will assume, as we do here, that our algorithm describes the case for $n \geq 2$ disks.

1. *Using the "two-step, three-in-a-row algorithm", recursively transport the $n-1$ smallest disks from post A to post C, using all three posts;*

2. *Move the largest disk from post A to post B in one move;*

3. *Transport the $n-1$ smallest disks from post C to post A, using all three posts.*

4. *Move the largest disk from post B to post C in one move;*

5. *Transport the $n-1$ smallest disks from post A to post C, using all three posts.*

Where $f(n)$ denotes the number of moves required according to the current algorithm, the number of moves required by each step above is as follows:

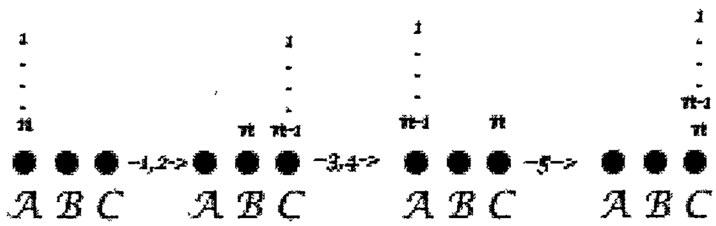

Figure 2: **Scorer, Grundy, and Smith's 'Two-step, Three-in-a-row' algorithm.**

- Step 1: $f(n-1)$,
- Step 2: 1,
- Step 3: $f(n-1)$,
- Step 4: 1,
- Step 5: $f(n-1)$.

$$\begin{cases} f(n) & = 3f(n-1)+2 \\ f(1) & = 2 \end{cases}$$

By induction, it is easily seen that $f(n) = 3^n - 1$.

Here we propose an algorithm for moving the tower of n disks one step from post A to post B:

Algorithm 3 : One-step, 3-in-a-row algorithm: (Also see Figure 3.)

1. *Recursively transport the n−1 smallest disks from post A to post C, using the two-step, 3-in-a-row algorithm;*
2. *Move the largest disk from post A to post B in one move;*
3. *Transport the n−1 smallest disks from post C to post B, using all three posts.*

Here is a diagram of our proposed algorithm for moving the tower one step from post A to post B. Indices indicate which disks are currently located on the indicated posts. Numbers embedded in arrows indicate which steps are being made. '1' is the smallest disk; n is the largest:

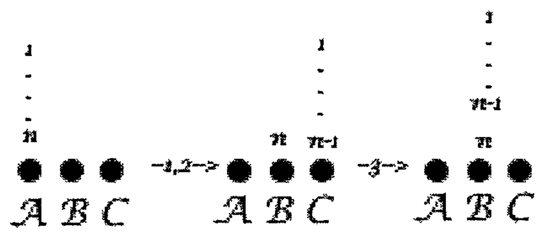

Figure 3: **One-step, 3-in-a-row algorithm**

The number of moves required by each step above is:

- Step 1: $3^{n-1} - 1$,
- Step 2: 1,
- Step 3: $f(n-1)$.

Thus we have obtained the recurrence relation:

$$\begin{cases} f(n) = f(n-1) + 3^{n-1} \\ f(1) = 1 \end{cases}$$

With respect to that relation, we see by induction:

$$\begin{aligned} f(n) &= f(n-1) + 3^{n-1} \\ &= f(n-2) + 3^{n-2} + 3^{n-1} \\ &= f(n-3) + 3^{n-3} + 3^{n-2} + 3^{n-1} \\ &= \cdots \\ &= f(1) + 3^1 + 3^2 + \cdots 3^{n-1} \\ &= 1 + 3 + 3^2 + \cdots + 3^{n-1} \\ &= \tfrac{3^n - 1}{2}. \end{aligned}$$

We will use this "one-step, three-in-a-row algorithm" later in a subsequent algorithm to move a tower of height n one step from post B to post C. Because this algorithm is designed to move the tower from post A to post B, we will justify here that moving the tower from post B to post C requires the same number of moves.

We move the tower from post B to post C as follows:

Algorithm 4 : One-step, 3-in-a-row algorithm (rev.)

1. *Recursively transport the $n-1$ smallest disks from post B to post A, using the two-step, 3-in-a-row algorithm;*

2. *Move the largest disk from post B to post C in one move;*

3. *Transport the $n-1$ smallest disks from post A to post C, using all three posts.*

The number of moves required by each step above is:

- Step 1: $f(n-1)$,

- Step 2: 1,

- Step 3: $3^{n-1}-1$.

Thus we have obtained the recurrence relation

$$\begin{cases} f(n) = f(n-1) + 3^{n-1} \\ f(1) = 1 \end{cases}$$

which matches the algorithm for moving the tower from post A to post B exactly.

Finally, we note that, on three posts, half the number of moves are required to transport n disks one step as it does to transport the same tower two steps.

CHAPTER 3

FOUR-IN-A-ROW PUZZLE

We will begin our study of the puzzle on four posts by first considering the history of that puzzle. We revisit "Frame's Conjecture" (1941) for solving the four-post puzzle in which disks may move from one post to any other as the tower is transported from their origin at post A to their destination at post D. Their algorithm is as follows:

Algorithm 5 : Frame's Conjecture

1. *Recursively transport a stack consisting of $n - i$ smallest disks from the first post to a temporary post, using all four posts in the process;*

2. *Transport the stack consisting of the i largest disks from the first post to the final post, using the standard three post algorithm and ignoring the post holding the smaller disks;*

3. *Recursively transport the smallest $n - i$ disks from the temporary post to the final post, again using all four posts in the process [3].*

"In addition, [Frame and Stewart] proved that if n is equal to the k-th triangular number $t_k = \frac{k(k+1)}{2}$, then the optimizing choice for i is in fact $i = k$, while if $t_{k-1} < n < t_k$ then both $k - 1$ and k are optimizing choices for i. Their proposed algorithm is in agreement with the partial solution of Dudeney" [8] given earlier and has been proven optimal [6]. Stockmeyer [8] has derived "a relatively simple exact closed form expression for the number ... of moves made by the Frame-Stewart algorithm." That expression is $O(\sqrt{n}2^{\sqrt{2n}})$. Frame's conjecture remains open for a puzzle on $n \geq 5$ posts.

In an article published in 1944, Scorer, Grundy, and Smith [7] propose a four-post variation, called the "four-in-a-row puzzle," in which the allowable moves are, in either direction, between posts A and B, posts B and C, posts C and D. This is the puzzle we consider in this chapter. Stockmeyer [8] proposes the following algorithm for moving the n disks from post A to post D, which he admits is "effective but not optimal":

Algorithm 6 : Stockmeyer's 'Four-in-a-row' Algorithm: (Also see Figure 4.)

1. *Recursively transport the stack consisting of the n−1 smallest disks from posts A to post D, using all four posts in the process;*
2. *Move the largest disk from post A to post C, in two moves;*
3. *Transport the smallest n−1 disks from D to post B using the 3-in-a-row algorithm, ignoring post A;*
4. *Move the largest disk from post C to post D;*
5. *Transport the smallest n−1 disks from post B to post D using the 3-in-a-row algorithm, again ignoring post A.*

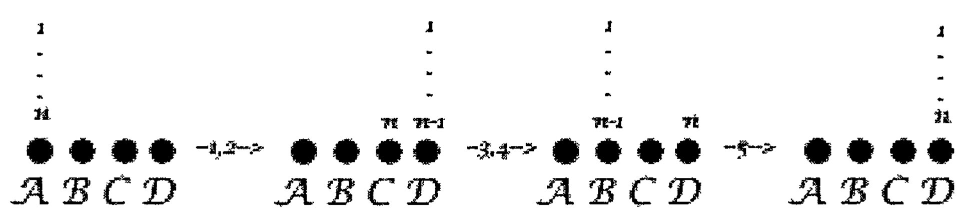

Figure 4: **Stockmeyer's 'Four-in-a-row' algorithm.**

The number of moves required by each step above is as follows:

- Step 1: $R(n-1)$,
- Step 2: 2,

- Step 3: $3^{n-1} - 1$,
- Step 4: 1,
- Step 5: $3^{n-1} - 1$.

Thus we obtain the recurrence relation:

$$\begin{cases} R(n) = R(n-1) + 2 \cdot 3^{n-1} + 1 \\ R(1) = 3 \end{cases}$$

whose solution, derived by induction, is

$$R(n) = 3^n + n - 1.$$

We propose the following algorithm and show that the algorithm requires only $O(4.7288^{\sqrt{n}})$ moves:

Algorithm 7 : 4-in-a-row algorithm: (Also see Figure 5.)

1. *Recursively transport k disks from post A to post D using all four posts in the process;*

2. *Transport $n-k-1$ disks from post A to post C, using the two-step, 3-in-a-row algorithm, ignoring post D; [We will soon see why it is now important to be more specific about this 3-in-a-row algorithm.]*

3. *Move the largest disk from post A to post B in one move;*

4. *Transport ℓ disks from post C to post A, using the two-step, 3-in-a-row algorithm, ignoring post D;*

5. *Transport k disks from post D to post A, using all four posts in the process;*

6. *Transport $n-k-\ell-1$ disks "one step" from post C to post D using the one step, 3-in-a-row algorithm;*

7. *Move the largest disk from post B to post C in one step;*

8. *Transport $n-k-\ell-1$ disks from post D to post B using the two-step, 3-in-a-row algorithm;*

9. *Move the largest disk from post C to post D in one step;*

10. *Transport $n-k-\ell-1$ disks from post B to post D using the two-step, 3-in-a-row algorithm;*

11. *Transport $k+\ell$ disks from post A to post D, using all four posts in the process.*

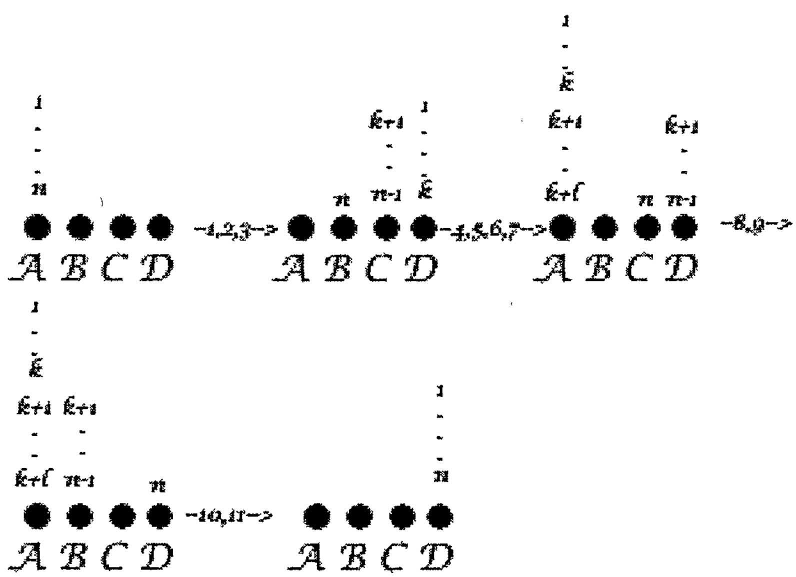

Figure 5: **4-in-a-row algorithm**

The number of moves required by each step above is as follows:

- Step 1: $f(k)$,
- Step 2: $3^{n-k-1}-1$,
- Step 3: 1,
- Step 4: $3^{\ell}-1$,

- Step 5: $f(k)$,
- Step 6: $\frac{1}{2}(3^{n-k-\ell-1}-1)$,
- Step 7: 1,
- Step 8: $3^{n-k-\ell-1}-1$,
- Step 9: 1,
- Step 10: $3^{n-k-\ell-1}-1$,
- Step 11: $f(k+\ell)$.

Thus we obtain the recurrence relation:

$$\begin{cases} f(n) = 2f(k) + f(k+\ell) + \frac{5}{2}(3^{n-k-\ell-1}-1) + 3^{n-k-1} + 3^{\ell} + 1 \\ f(1) = 3 \end{cases}$$

For n from 1 to 6, the number of moves required is 3, 10, 19, 34, 57, and 88, [2] and this algorithm matches those numbers of moves for best values of k and ℓ. However, because two variables k and ℓ are involved, this recurrence relation turns out to be very difficult to solve. So we reduce the algorithm to one that is more readily solved by setting $\ell = 0$. This revised algorithm is still valid for all k with $1 \leq k \leq n-1$. So in order to minimize the number of moves for this revised algorithm, we choose k which minimizes the function $3f(k) + \frac{7}{2} \cdot 3^{n-k-1} - \frac{1}{2}$. We obtain the recurrence relation:

$$f(n) = \min_{k} \left\{ 3f(k) + \frac{7}{2} \cdot 3^{n-k-1} - \frac{1}{2} \right\} \tag{1}$$

We use *Mathematica* [1] to evaluate this relation for values of n, and k up to $n = 5000$. We notice that $k = n - \lfloor \sqrt{2n} + .5 \rfloor$ can always minimize the above objective function.

[1]Complete output is available upon request. A summary of the output is displayed in Appendix B.

Conjecture 1 $f(n) = 3f\left(n - \lfloor\sqrt{2n} + .5\rfloor\right) + \frac{7}{2}(3^{\lfloor\sqrt{2n}+.5\rfloor-1} - \frac{1}{2}$ *with* $f(1) = 3$

Although we cannot prove this conjecture, we can use induction to prove the following:

Theorem 1

For any sufficiently small $\varepsilon > 0, f(n) \leq c \cdot (3+\varepsilon)^{\sqrt{2n}}$, *where* $c = \frac{7(3+\varepsilon)}{6\varepsilon}$.

Proof. *Base Case*: Since $f(1) = 3$, *Theorem 1* holds for $n = 1$ and for all sufficiently small $\varepsilon > 0$.

Now suppose $n \geq 2$. Set $k = n - \sqrt{2n}$ and suppose $f(k) \leq c \cdot (3+\varepsilon)^{\sqrt{2k}}$. Note that $\sqrt{2(n - \sqrt{2n})} \leq \sqrt{2(\sqrt{n} - \frac{\sqrt{2}}{2})^2} = \sqrt{2}(\sqrt{n} - \frac{\sqrt{2}}{2}) = \sqrt{2n} - 1$.
By (1) above,

$$\begin{aligned}
f(n) &\leq 3f(k) + \tfrac{7}{2} \cdot 3^{n-k-1} - \tfrac{1}{2} \\
&\leq 3c \cdot (3+\varepsilon)^{\sqrt{2(n-\sqrt{2n})}} + \tfrac{7}{6} \cdot 3^{\sqrt{2n}} - \tfrac{1}{2} \\
&\leq 3c \cdot (3+\varepsilon)^{\sqrt{2n}-1} + \tfrac{7}{6} \cdot 3^{\sqrt{2n}} - \tfrac{1}{2} \\
&\leq c \cdot (3+\varepsilon)^{\sqrt{2n}} + \tfrac{7}{6} \cdot 3^{\sqrt{2n}} - \tfrac{\varepsilon}{3+\varepsilon} \cdot c \cdot (3+\varepsilon)^{\sqrt{2n}} \\
&\leq c \cdot (3+\varepsilon)^{\sqrt{2n}} + \tfrac{7}{6} \cdot 3^{\sqrt{2n}} - \tfrac{\varepsilon}{3+\varepsilon} \cdot \tfrac{7(3+\varepsilon)}{6\varepsilon} \cdot 3^{\sqrt{2n}} \\
&= c \cdot (3+\varepsilon)^{\sqrt{2n}}
\end{aligned}$$

□

Since $3^{\sqrt{2}} < 4.72881$, we have

$$\begin{aligned}
f(n) &\leq \tfrac{7(3+\varepsilon)}{6\epsilon} \cdot (3+\varepsilon)^{\sqrt{2n}} \\
&= O\left((3+\varepsilon)^{\sqrt{2n}}\right) \\
&= O\left(4.72881^{\sqrt{n}}\right)
\end{aligned}$$

Finally, we note the order of this result, $4.72881^{\sqrt{n}}$, in comparison with the order of the proposed result, 3^n, proposed by Stockmeyer in 1994 [8].

CHAPTER 4

FOUR-POST CYCLIC PUZZLE

The 4-post cyclic puzzle, a variation of the four post puzzle, was proposed by Scorer, Grundy, and Smith [7]. The restrictions permit that disks may only be moved clockwise along a directed cycle, from post A to post B, from B to C, and from D to A. The authors propose the following algorithm to accomplish the task of transporting a tower of n disks (where 'disk 1' is the smallest and ' disk n' is the largest) two steps along the cycle, say from post A to post C:

Algorithm 8 : Scorer, Grundy and Smith's 'Four-post cyclic' algorithm: (Also see Figure 6.)

1. *Recursively transport the stack consisting of the n−1 smallest disks from post A to post C;*

2. *Move the largest disk from post A to post B;*

3. *Transport the n−1 smallest disks from post C to post A;*

4. *Move the largest disk from post B to post C;*

5. *Transport the n−1 smallest disks from post A to post C.*

Letting $N(n)$ denote the number of moves made by this algorithm for a stack of n disks, the number of moves required by each step above is as follows:

- Step 1: $N(n-1)$,

- Step 2: 1,

- Step 3: $N(n-1)$,

- Step 4: 1,

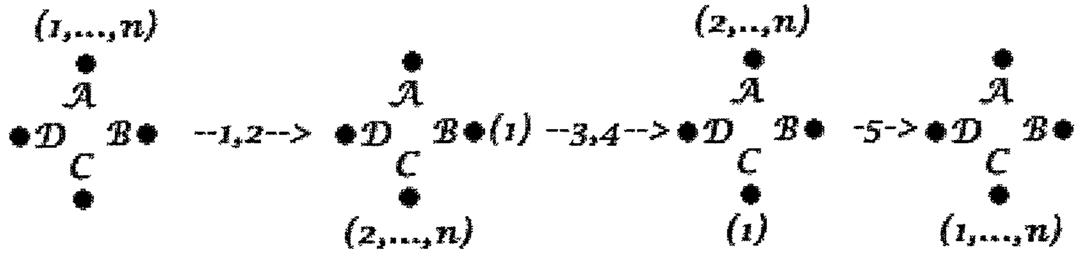

Figure 6: **Scorer, Grundy and Smith's 'Four-post cyclic' algorithm**

- Step 5: $N(n-1)$.

Thus we obtain the recurrence relation:

$$\begin{cases} N(n) = 3N(n-1) + 2 \\ N(1) = 2 \end{cases}$$

which is equivalent to the "two-step, 3-in-a-row algorithm" whose solution was presented earlier [2] and is

$$N(n) = 3^n - 1$$

We propose a three-phase algorithm for moving the tower three steps from post A to post D. To transport the tower from post A to post C, as Scorer, Grundy, and Smith's algorithm does, simply apply this algorithm twice. Similarly, to transport the tower to post B, apply the algorithm three times. Our proposal is as follows:

- *Phase 1: We split the tower of height $n = 2k$, for convenience, ['disk 1' is the smallest, 'disk $2k$' is the largest] into two towers on two posts: odd-numbered disks 1 through $2k-1$, and even-numbered disks 2 through $2k$, to posts D and C, respectively. We will call the tower of odd-numbered disks the "odd tower" and the tower of even-numbered disks the "even tower."*

- *Phase 2: Simultaneously, recursively transport the two towers.*

- *Phase 3: Reassemble the two towers onto post D.*

We must note that this algorithm is designed to transport a tower consisting of an even number of disks. It is a small adjustment to alter the algorithm for an odd number of posts and for large values of n, this will not alter the O approximation for the minimum number of moves.

We accomplish Phase 1, the splitting of the towers, in the following way:

Algorithm 9 : Phase 1: Splitting the tower (Also see Figure 7.)

1. *We leave disks n and $n-1$ on post A and split the tower of height $n-2$ into two towers, one containing odd-numbered disks and the other containing even-numbered disks, each of height $\frac{n}{2}-1$ onto posts D and C, respectively;*

2. *Move the second largest disk one step to post B;*

3. *Simultaneously, recursively transport the "even" and "odd" towers two steps, the "even" tower moving from post C to post A and the "odd" tower from post D to post B; [This algorithm is "Phase 2". It is described below.]*

4. *Simultaneously, recursively transport the "odd" and "even" towers, each of height $\frac{n}{2}$, two steps, the "even" tower moving from post A to post C and the "odd" tower from post B to post D.*

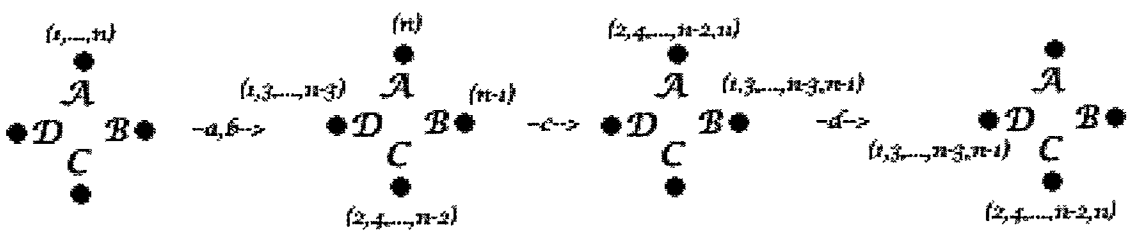

Figure 7: **Phase 1: Splitting the tower**

Where $g(n)$ indicates the number of moves required to split a tower of height n into two towers, each of height $\frac{n}{2}$, onto two posts, and where $f(n, i)$ indicates the number of moves required to simultaneously transport two towers i steps, each of height $\frac{n}{2}$, the number of moves required by each step above is as follows:

- Step 1: $g(n-2)$,
- Step 2: 1,
- Step 3: $f(\frac{n}{2}-1,2)$,
- Step 4: 1,
- Step 5: $f(\frac{n}{2},2)$.

Thus we obtain the following recurrence relation:

$$g(n) \leq g(n-2) + 1 + f\left(\frac{n}{2}-1,2\right) + f\left(\frac{n}{2},2\right)$$

We will solve this relation following the summary of the three phases.

We accomplish Phase 3, the reassembling of the two towers, simply by reversing the algorithm in Phase 1. We accomplish Phase 2, the simultaneous transport of two towers, in the following way:

Algorithm 10 : Phase 2: Simultaneous transport of towers one step: (Also see Figure 8.)

1. *We leave disk $2k$ on post C and disk $2k-1$ on post D and simultaneously, recursively transport the two towers three steps, odds to post C and evens to post B;*
2. *Move disk $2k-1$ one step from post D to the goal post at A;*
3. *Simultaneously transport the two towers three steps, odds to post B and evens to post A;*
4. *Move disk $2k$ one step from post C to the goal post at D;*

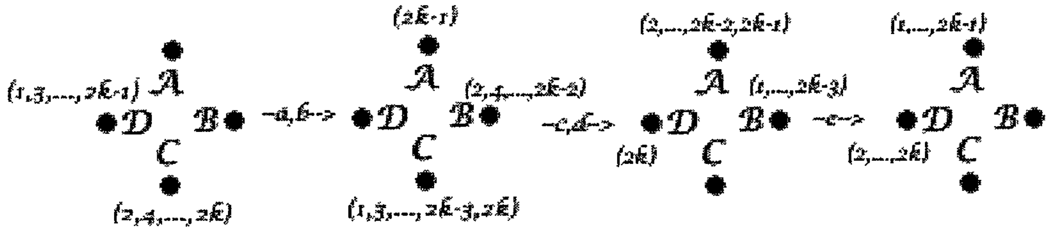

Figure 8: **Phase 2: Simultaneous transport of towers one step**

5. *Simultaneously transport the two towers three steps, odds to their goal post at A and evens to their goal post at D.*

Where $f(k, i)$ indicates a simultaneous transport of two towers, each of height k, i steps, the number of moves required by each step above is as follows:

- Step 1: $f(k-1, 3)$,
- Step 2: 1,
- Step 3: $f(k-1, 3)$,
- Step 4: 1,
- Step 5: $f(k-1, 3)$,

Thus we obtain the following recurrence relation:

$$f(k, 1) = 3f(k-1, 3) + 2.$$

Algorithm 11 : Phase 2: Simultaneous transport of towers two steps: (Also see Figure 9.)

1. *We leave disk $2k$ on post C and disk $2k-1$ on post D and simultaneously, recursively transport the two towers three steps, odds to post C and evens to post B;*

2. Move disk $2k-1$ *one step from post D to post A;*

3. Simultaneously transport the two towers one step, odds to post D and evens to post C;

4. Move disk $2k-1$ *one step from post A to the goal post at B;*

5. Simultaneously transport the two towers two steps, odds to post B and evens to post A;

6. Move disk $2k$ *one step from post C to post D;*

7. Simultaneously transport the two towers one step, odds to post C and evens to post B;

8. Move disk $2k$ *one step from post D to the goal post at A;*

9. Simultaneously transport the two towers three steps, odds to their goal post at B and evens to their goal post at A.

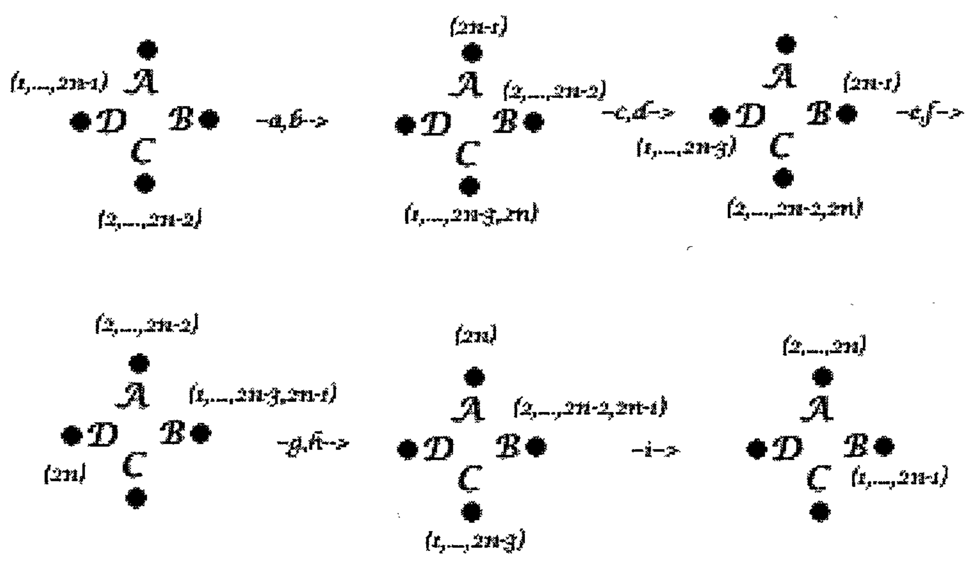

Figure 9: **Phase 2: Simultaneous transport of towers two steps**

Where $f(k, i)$ indicates a simultaneous transport of two towers, each of height k, i steps, the number of moves required by each step above is as follows:

- Step 1: $f(k-1, 3)$,
- Step 2: 1,
- Step 3: $f(k-1, 1)$,
- Step 4: 1,
- Step 5: $f(k-1, 2)$,
- Step 6: 1,
- Step 7: $f(k-1, 1)$,
- Step 8: 1,
- Step 9: $f(k-1, 3)$.

Thus we obtain the following recurrence relation:

$$f(k, 2) = 2f(k-1, 1) + f(k-1, 2) + 2f(k-1, 3) + 4$$

Algorithm 12 : Phase 2: Simultaneous transport of towers three steps: (Also see Figure 10.)

1. *We leave disk $2k$ on post C and disk $2k-1$ on post D and simultaneously, recursively transport the two towers three steps, odds to post C and evens to post B;*
2. *Move disk $2k-1$ one step from post D to post A;*
3. *Simultaneously transport the two towers one step, odds to post D and evens to post C;*
4. *Move disk $2k-1$ one step from post A to post B;*

5. *Simultaneously transport the two towers two steps, odds to post B and evens to post A;*
6. *Move disk* $2k$ *one step from post C to post D;*
7. *Simultaneously transport the two towers one step, odds to post C and evens to post B;*
8. *Move disk* $2k$ *one step from post D to post A;*
9. *Simultaneously transport the two towers two steps, odds to post A and evens to post D;*
10. *Move disk* $2k-1$ *one step from post B to its goal post at C;*
11. *Simultaneously transport the two towers three steps, odds to post D and evens to post C;*
12. *Move disk* $2k$ *one step from post A to its goal post at B;*
13. *Simultaneously transport the two towers three steps, odds to their goal post at C and evens to their goal post at B.*

Where $f(k,i)$ indicates a simultaneous transport of two towers, each of height k, i steps, the number of moves required by each step above is as follows:

- Step 1: $f(k-1,3)$,
- Step 2: 1,
- Step 3: $f(k-1,1)$,
- Step 4: 1,
- Step 5: $f(k-1,2)$,
- Step 6: 1,
- Step 7: $f(k-1,1)$,

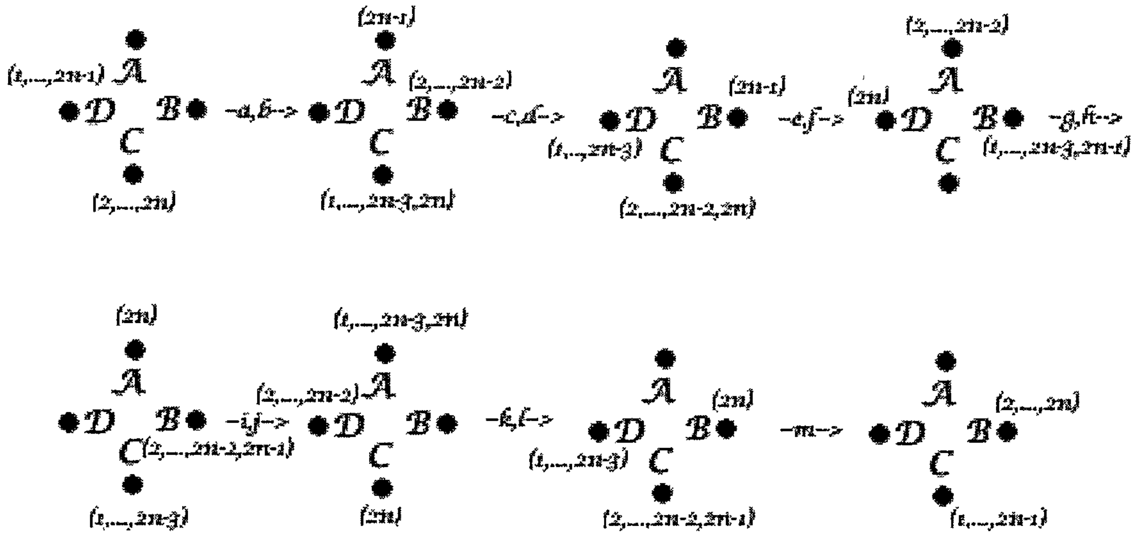

Figure 10: **Phase 2: Simultaneous transport of towers three steps**

- Step 8: 1,
- Step 9: $f(k-1,2)$,
- Step 10: 1,
- Step 11: $f(k-1,3)$,
- Step 12: 1,
- Step 13: $f(k-1,3)$.

Thus we obtain the following recurrence relation:

$$f(k,3) = 2f(k-1,1) + 2f(k-1,2) + 3f(k-1,3) + 6$$

In summary, where $f(k,i)$ indicates a simultaneous transport of two towers, each of height k, $\imath$ steps, we obtain the recurrence relations:

$$f(k,1) = 3f(k-1,3) + 2$$
$$f(k,2) = 2f(k-1,1) + f(k-1,2) + 2f(k-1,3) + 4$$
$$f(k,3) = 2f(k-1,1) + 2f(k-1,2) + 3f(k-1,3) + 6$$

We endeavor to solve the system of recurrence relations with a matrix equation as follows:

$$\begin{pmatrix} f(k,1) \\ f(k,2) \\ f(k,3) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} f(k-1,1) \\ f(k-1,2) \\ f(k-1,3) \end{pmatrix} + \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}$$

We simplify matters by finding a system that does not involve the constant matrix on the right as follows:

$$\begin{pmatrix} f(k,1)+x \\ f(k,2)+y \\ f(k,3)+z \end{pmatrix} = \begin{pmatrix} 0 & 0 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} f(k-1,1)+x \\ f(k-1,2)+y \\ f(k-1,3)+z \end{pmatrix}$$

We compute the values of $x, y,$ and z. Computing x we get

$$f(k,1) + x = 3f(k-1,1) + 3z \qquad (2)$$
$$f(k,1) = 3f(k-1,1) + 2 \qquad (3)$$

We subtract (3) from (2) and hence

$$x = 3z - 2$$

which implies

$$x - 3z = -2.$$

Computing y we get

$$\begin{aligned} f(k,2) &= 2f(k-1,1)+f(k-1,2)+2f(k-1,3)+4 && (4)\\ f(k,2)+y &= 2f(k-1,1)+2x+f(k-1,2)+y+2f(k-1,3)+2z && (5) \end{aligned}$$

We subtract (4) from (5) and hence

$$y = 2x + y + 2z - 4$$

which implies

$$2x + 2z = 4.$$

Computing z we get

$$\begin{aligned} f(k,3) &= 2f(k-1,1)+2f(k-1,2)+3f(k-1,3)+6 && (6)\\ f(k,3)+z &= 2f(k-1,1)+2x+2f(k-1,1)+2y+3f(k-1,3)+3z && (7) \end{aligned}$$

We subtract (6) from (7) and hence

$$z = 2x + 2y + 3z - 6$$

which implies

$$2x + 2y + 2z = 6.$$

We are left with the following system:

$$\begin{cases} x - 3z & = -2 \\ 2x + 2z & = 4 \\ 2x + 2y + 2z & = 6 \end{cases}$$

whose solution is $(x, y, z) = (1, 1, 1)$.

We obtain the following modified system :

$$\begin{pmatrix} f(k,1)+1 \\ f(k,2)+1 \\ f(k,3)+1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} f(k-1,1)+1 \\ f(k-1,2)+1 \\ f(k-1,3)+1 \end{pmatrix}$$

With respect to the 3×3 matrix above, call it A, we find a diagonal matrix D of eigenvalues $\lambda_1,\ \lambda_2,$ and λ_3 and a matrix X' of corresponding eigenvectors Z, X, and Y that satisfies the equation $AX' = \lambda X'$. The eigenvalues $\lambda_1,\ \lambda_2,$ and λ_3 of A are given in the diagonal matrix

$$\begin{pmatrix} 5.47783 & 0 & 0 \\ 0 & -0.738917 + 0.741165i & 0 \\ 0 & 0 & -0.738917 - 0.741165i \end{pmatrix}$$

and the corresponding eigenvectors Z, X, and Y of A are given [2] in the column matrix

$$\begin{bmatrix} Z & X & Y \end{bmatrix} = \begin{pmatrix} 0.547762 & -2.02383 - 2.02999i & -2.02383 + 2.02999i \\ 0.691255 & 0.154373 + 2.40057\imath & 0.154373 - 2.40057i \\ 1 & 1 & 1 \end{pmatrix}.$$

We may express complex eigenvalues and eigenvectors in the following way: Write $X = x + \imath\, Y$and $\lambda = a + b\imath$. Then

$$\begin{aligned} Ax + A\imath\, Y &= A(x + \imath\, Y) = (a + b\imath)(x + \imath\, Y) \\ &= ax + ai\, Y + b\imath\, x - bY \\ &= (ax - bY) + \imath\, (aY + bx) \end{aligned}$$

We identify the real parts as $AX = ax - bY$ and the imaginary parts as $AY = aY + bx = aY + bx$.

[2] All calculations herein are performed using *Mathematica* and may be found in Appendix A.

$$
\begin{aligned}
A\begin{bmatrix} Z & X & Y \end{bmatrix} &= \begin{bmatrix} AZ & AX & AY \end{bmatrix} \\
&= \begin{bmatrix} AZ & (ax-bY) & (aY+bx) \end{bmatrix} \\
&= \begin{pmatrix} 3 & 3 & 0 \\ 3.787 & -1.893 & -1.66 \\ 5.478 & -0.739 & 0.741 \end{pmatrix} \\
&= \begin{bmatrix} Z & X & Y \end{bmatrix} \begin{pmatrix} 5.478 & 0 & 0 \\ 0 & -0.739 & 0.741 \\ 0 & -0.741 & 0.739 \end{pmatrix}
\end{aligned}
$$

Let D denote the matrix above on the right. Then

$$
A = \begin{bmatrix} Z & X & Y \end{bmatrix} \cdot D \cdot \begin{bmatrix} Z & X & Y \end{bmatrix}^{-1}
$$

Thus

$$
A^{k-1} = \begin{bmatrix} Z & X & Y \end{bmatrix} \cdot D^{k-1} \cdot \begin{bmatrix} Z & X & Y \end{bmatrix}^{-1}
$$

Now we factor out $\lambda_1 = 5.478$; then

$$
\begin{aligned}
D &= \lambda_1 \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{-0.739}{\lambda_1} & \frac{0.741}{\lambda_1} \\ 0 & \frac{-0.741}{\lambda_1} & \frac{0.739}{\lambda_1} \end{pmatrix} \\
D^{k-1} &= \lambda_1^{k-1} \cdot \left(\begin{array}{c|c} 1 & 0 \quad 0 \\ \hline 0 & \\ & N \\ 0 & \end{array}\right)
\end{aligned}
$$

where

$$
N = \begin{pmatrix} \frac{-0.739}{\lambda_1} & \frac{0.741}{\lambda_1} \\ \frac{-0.741}{\lambda_1} & \frac{0.739}{\lambda_1} \end{pmatrix}^{k-1} \approx \begin{pmatrix} -.135 & -.135 \\ -.135 & -.135 \end{pmatrix}^{k-1}
$$

So

$$D^{k-1} \approx \lambda_1^{k-1} \cdot \left(\begin{array}{c|c} 1 & 0 \quad 0 \\ \hline 0 & \\ 0 & \begin{pmatrix} -.135 & -.135 \\ -.135 & -.135 \end{pmatrix}^{k-1} \end{array} \right)$$

Then we have

$$A^{k-1} \approx \lambda_1^{k-1} \cdot \begin{bmatrix} Z & X & Y \end{bmatrix} \left(\begin{array}{c|c} 1 & 0 \quad 0 \\ \hline 0 & \\ 0 & \begin{pmatrix} -.135 & -.135 \\ -.135 & -.135 \end{pmatrix}^{k-1} \end{array} \right) \begin{bmatrix} Z & X & Y \end{bmatrix}^{-1}$$

As $k \to \infty$, the entries in the 3×3 matrix above become arbitrarily small. So we may represent that matrix as some constant matrix as follows:

$$A^{k-1} \approx \lambda_1^{k-1} \cdot \begin{bmatrix} Z & X & Y \end{bmatrix} \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \begin{bmatrix} Z & X & Y \end{bmatrix}^{-1}$$

Since each of the matrices on the right is independent of k, we may represent their product as follows:

$$A^{k-1} \approx \lambda_1^{k-1} \cdot \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Recall the system we endeavor to solve:

$$
\begin{aligned}
\begin{pmatrix} f(k,1)+1 \\ f(k,2)+1 \\ f(k,3)+1 \end{pmatrix} &= A \cdot \begin{pmatrix} f(k-1,1)+1 \\ f(k-1,2)+1 \\ f(k-1,3)+1 \end{pmatrix} \\
&= A \cdot A \cdot \begin{pmatrix} f(k-2,1)+1 \\ f(k-2,2)+1 \\ f(k-2,3)+1 \end{pmatrix} \\
&\quad . \\
&\quad . \\
&\quad . \\
&= A^{k-1} \begin{pmatrix} f(1,1)+1 \\ f(1,2)+1 \\ f(1,3)+1 \end{pmatrix} \\
&= \lambda_1^{k-1} \cdot \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \begin{pmatrix} f(1,1)+1 \\ f(1,2)+1 \\ f(1,3)+1 \end{pmatrix} \\
&= \lambda_1^{k-1} \cdot \begin{bmatrix} c_{1'} \\ c_{2'} \\ c_{3'} \end{bmatrix}
\end{aligned}
$$

Also recall that we chose $n = 2k$. Therefore

$$
\begin{aligned}
f(k,1) &= \lambda_1^{k-1} \cdot c_{1'} \\
&= \lambda_1^{\frac{n}{2}-1} \cdot c_{1'} \\
&= \lambda_1^{\frac{n}{2}} \cdot c_{1''} \\
&= (\sqrt{\lambda_1})^n \cdot c_1'' \\
&= (\sqrt{5.478})^n \cdot c_1'' \\
&\approx c_1'' \cdot 2.34^n
\end{aligned}
$$

Finally, the number of moves required to complete Phase 2 is $O(2.34^n)$. And now we will solve the system used in Phases 1 and 3. Recall the associated recurrence relation:

$$\begin{aligned}
g(n) &\leq g(n-2) + 1 + f(\tfrac{n}{2} - 1, 2) + f(\tfrac{n}{2}, 2) \\
&\leq g(n-2) + 1 + c_0 \cdot 2.34^n + c_1 \cdot 2.34^n \\
&\leq g(n-2) + c_2 \cdot 2.34^n \\
g(n-2) &\leq g(n-4) + c_2 \cdot 2.34^{n-2} \\
g(n-4) &\leq g(n-6) + c_2 \cdot 2.34^{n-4} \\
&\vdots \\
g(4) &\leq g(2) + c_2 \cdot 2.34^{n-4}
\end{aligned}$$

We sum these and get

$$\begin{aligned}
g(n) &\leq g(2) + c_2 \cdot [2.34^n + 2.34^{n-2} + \cdots + 2.34^4] \\
&\leq \tfrac{c\,2\,34^n}{1 - \frac{1}{2\,34}} \\
&\approx c' \cdot 2.34^n.
\end{aligned}$$

In summary, each of the three Phases of the algorithm requires on the order of $(2.34)^n$ moves. Thus, the number of moves required is of order 2.34^n. As we mentioned earlier, to transport the tower to post C instead of post D, apply the algorithm twice. To transport the tower to post B instead of post D, apply the algorithm three times. Thus, our estimate for the minimum number of moves is of the same order regardless of the tower's destination.

We compare the order of this result, 2.34^n, with that proposed in the article [7], 3^n.

Contained herein are the *Mathematica* calculations used in Chapter Four.

Matrix A is our coefficient matrix pertaining to the three recurrence relations we developed for use in each of the three Phases of our algorithm:

```
A = MatrixForm[{{0, 0, 3}, {2, 1, 2}, {2, 2, 3}}]
```

$$\begin{pmatrix} 0 & 0 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{pmatrix}$$

$$\texttt{N[Eigensystem[A =} \begin{pmatrix} 0 & 0 & 3 \\ 2 & 1 & 2 \\ 2 & 2 & 3 \end{pmatrix} \texttt{], 3]}$$

```
{{5.47783, -0.738917 + 0.741165 i, -0.738917 - 0.741165 i},
 {{0.547662, 0.691255, 1.}, {-2.02383 - 2.02999 i, 0.154373 + 2.40057 i, 1.},
  {-2.02383 + 2.02999 i, 0.154373 - 2.40057 i, 1.}}}
```

This '**Eigensystem**' is a matrix consisting of A's eigenvalues and eigenvectors:

```
N[{evalues, evectors} = Eigensystem[A], 3]
```

```
{{5.47783, -0.738917 + 0.741165 i, -0.738917 - 0.741165 i},
 {{0.547662, 0.691255, 1.}, {-2.02383 - 2.02999 i, 0.154373 + 2.40057 i, 1.},
  {-2.02383 + 2.02999 i, 0.154373 - 2.40057 i, 1.}}}
```

Matrix **d** is a diagonal matrix of eigenvalues:

```
d = N[DiagonalMatrix[evalues] // MatrixForm, 3]
```

$$\begin{pmatrix} 5.47783 & 0. & 0. \\ 0. & -0.738917 + 0.741165\, i & 0. \\ 0. & 0. & -0.738917 - 0.741165\, i \end{pmatrix}$$

```
P = Transpose[evectors];
```

```
N[P, 3] // MatrixForm
```

$$\begin{pmatrix} 0.547662 & -2.02383 - 2.02999\, i & -2.02383 + 2.02999\, i \\ 0.691255 & 0.154373 + 2.40057\, i & 0.154373 - 2.40057\, i \\ 1. & 1. & 1. \end{pmatrix}$$

```
N[Inverse[P], 3] // MatrixForm
```

$$\begin{pmatrix} 0.330524 - 1.01049\times 10^{-17}\, i & 0.279501 - 8.54499\times 10^{-18}\, i & 0.625778 - 1.91315\times 10^{-17}\, i \\ -0.165262 + 0.0369605\, i & -0.13975 - 0.177029\, i & 0.187111 + 0.10213\, i \\ -0.165262 - 0.0369605\, i & -0.13975 + 0.177029\, i & 0.187111 - 0.10213\, i \end{pmatrix}$$

The following matrices are useful in solving the matrix equation we desired:

```
AZ = MatrixForm[{{0, 0, 3}, {2, 1, 2}, {2, 2, 3}}.{0.548, 0.691, 1}]
```

$$\begin{pmatrix} 3. \\ 3.787 \\ 5.478 \end{pmatrix}$$

```
ax = MatrixForm[-0.739 {-2.024, 0.154, 1}]
```

$$\begin{pmatrix} 1.49574 \\ -0.113806 \\ -0.739 \end{pmatrix}$$

```
aY = MatrixForm[-0.739 {-2.03, 2.401, 0}]
```

$$\begin{pmatrix} 1.50017 \\ -1.77434 \\ 0 \end{pmatrix}$$

```
bx = MatrixForm[0.741 {-2.024, 0.154, 1}]
```

$$\begin{pmatrix} -1.49978 \\ 0.114114 \\ 0.741 \end{pmatrix}$$

```
bY = MatrixForm[0.741 {-2.03, 2.401, 0}]
```

$$\begin{pmatrix} -1.50423 \\ 1.77914 \\ 0 \end{pmatrix}$$

ax - bY:

$$\mathbf{MatrixForm}\left[\begin{pmatrix} 1.495736\text{`} \\ -0.11380599999999999\text{`} \\ -0.739\text{`} \end{pmatrix} - \begin{pmatrix} -1.5042299999999997\text{`} \\ 1.7791409999999999\text{`} \\ 0 \end{pmatrix}\right]$$

$$\begin{pmatrix} 2.99997 \\ -1.89295 \\ -0.739 \end{pmatrix}$$

aY + bx:

$$\mathbf{MatrixForm}\left[\begin{pmatrix} 1.5001699999999998\text{`} \\ -1.7743389999999999\text{`} \\ 0 \end{pmatrix} + \begin{pmatrix} -1.499784\text{`} \\ 0.114114\text{`} \\ 0.741\text{`} \end{pmatrix}\right]$$

$$\begin{pmatrix} 0.000386 \\ -1.66022 \\ 0.741 \end{pmatrix}$$

$$\mathbf{ZXY = MatrixForm}\left[\mathbf{Transpose}\left[\begin{pmatrix} 0.548\text{`} & 0.691\text{`} & 1 \\ -2.024\text{`} & 0.154\text{`} & 1 \\ -2.03\text{`} & 2.401\text{`} & 0 \end{pmatrix}\right]\right]$$

$$\begin{pmatrix} 0.548 & -2.024 & -2.03 \\ 0.691 & 0.154 & 2.401 \\ 1 & 1 & 0 \end{pmatrix}$$

$[Z\,X\,Y]^{-1}$:

```
N[Inverse[( 0.548`  -2.024`  -2.03`
            0.691`   0.154`   2.401`
            1        1        0      )], 3] // MatrixForm
```

$$\begin{pmatrix} 0.330467 & 0.279403 & 0.625837 \\ -0.330467 & -0.279403 & 0.374163 \\ -0.0739111 & 0.354003 & -0.204113 \end{pmatrix}$$

```
p = MatrixForm[{{5.478, 0, 0}, {0, -.739, .741}, {0, -.741, -.739}}]
```

$$\begin{pmatrix} 5.478 & 0 & 0 \\ 0 & -0.739 & 0.741 \\ 0 & -0.741 & -0.739 \end{pmatrix}$$

Contained herein is a summary of output detailing the number of moves required by our proposed algorithm in Chapter Three.

```
f[n_, k_] := 3 f[k] + 7 / 2 (3^(n - k - 1)) - 1 / 2;
f[0] = 0;
f[1] = 3;
f[2] = 10;

f[n_] := f[n] = Min[Flatten[Table[f[n, k], {k, 0, n - 2}]]]

T[n_] := Take[Position[Table[f[n, k], {k, 0, n - 2}], f[n]], 1] - {{1}}

TableForm[Table[{n, T[n], f[n], n - T[n], Floor[√(2 n) + .5]}, {n, 3, 5000}]]
```

3	1	19	2	2
4	1	40	3	3
5	2	61	3	3
6	3	88	3	3
7	3	151	4	4
8	4	214	4	4
9	5	277	4	4
10	6	358	4	4
100	86	57262768	14	14
200	180	52592331382	20	20
300	276	7837419637348	24	24
400	372	629747258968510	28	28
500	468	27949721181848602	32	32
600	565	851926031824633900	35	35
700	663	18173958757211643568	37	37
800	760	372497361148327568542	40	40
900	858	5054549414602918149016	42	42
1000	955	73283367970830402580228	45	45
1100	1053	866352201196031982127060	47	47
1200	1151	8913900013840502977230592	49	49
1300	1249	83575290733793172209336488	51	51
1400	1347	722025901121080741155677932	53	53
1500	1445	5684136792047165834735639020	55	55
1600	1543	38945786357786079977638815688	57	57
1700	1642	276196713330189523514430705208	58	58
1800	1740	1958236005880126290118656733702	60	60
1900	1838	11689361894390868119066689955176	62	62
2000	1937	69341337148684904898507119108536	63	63
2100	2035	415761882573751720017321227227108	65	65
2200	2134	2100556077063944512489439180367652	66	66
2300	2232	11982698111865795443334728859318370	68	68
2400	2331	57039497201744144267015024608863652	69	69
2500	2429	291409020039629876419821172046827588	71	71

2600	2528	1391128671899994019931129131267210582	72	72
2700	2627	5671524585438437158180757227474012372	73	73
2800	2725	29281287308534691941789232200235740140	75	75
2900	2824	124035735794548859517988894161529468186	76	76
3000	2923	476424961476487425667992487218723118708	77	77
3100	3021	2218501041487543904586767442079141901956	79	79
3200	3120	9127195216335860764768138230244168626862	80	80
3300	3219	34451774655993795626257061578937840282500	81	81
3400	3318	123531229183039605411416542929984840970036	82	82
3500	3416	515815862448713685892301382817726112803090	84	84
3600	3515	2008389421874778819538109639481784651977988	85	85
3700	3614	7324186162932315605677726211344153566900352	86	86
3800	3713	25618189361887081842662712972167801544380920	87	87
3900	3812	87037192248430243358194769309226253196524450	88	88
4000	3911	289396643864093501825158281240797783376744712	89	89
4100	4009	1016080425384877854914190189902415674503158688	91	91
4200	4108	3628650853432466441064348502092449391622232422	92	92
4300	4207	12443894057201055991214659956363944005064095912	93	93
4400	4306	41455644957524612094877665178144585683361872388	94	94
4500	4405	135089238115969590999840577337970916587326537260	95	95
4600	4504	432485868734990304199664055053660923971898870486	96	96
4700	4603	1364265803336898307362977867167637381340043882828	97	97
4800	4702	4248683203318525509871566906185106232676941948348	98	98
4900	4801	13080090595615213503434629228588591745152562201896	99	99
5000	4900	39841545561032987063179759003825666720276812048022	100	100

BIBLIOGRAPHY

1. Dudeney, H. E., *The Canterbury Puzzles.* New York: Dover Publications, 2002.

2. Er, M. C., "The Cyclic Towers of Hanoi: A Representation Approach," *The Computer Journal* 27 (1984): 171-175.

3. Frame, J. S., Stewart, B. M., "3918," *The American Mathematical Monthly* 48, No. 3. (1941): 216-219.

4. Klavžar, S., Milutinović, U., and Petr, C., *On the Frame-Stewart algorithm for the multi-peg Tower of Hanoi problem*, Extended Abstract, Department of Mathematics, PEF, University of Maribor, 2000.

5. Lawrence Hall of Science: University of California, Berkeley, *Tower of Hanoi.* http://www.lhs.berkeley.edu/Java/Tower/towerhistory.html.

6. Majumdar, A. A. K. "Frame's Conjecture and the Tower of Hanoi problem with four pegs. (English. English summary)," *Indian Journal of Mathematics* 36 (1994): 215-217.

7. Scorer, R. S., Grundy, P.M., and Smith, C. A. B., "Some Binary Games," *The Mathematical Gazette* 280 (1944): 96-103.

8. Stockmeyer, P. K.,"Variations on the Four-Post Tower of Hanoi Puzzle," *Congressus Numerantium* 102 (1994): 3-12.

VITA

Steven Greenstein was born in Silver Spring, Maryland, on September 3, 1969, to parents Leonard Greenstein and Linda Goldberg Greenstein. After completing his work at North Springs High School in Atlanta, Georgia, in 1988, he entered Georgia Institute of Technology. Upon deciding to include teaching in his coursework, he transferred to Georgia State University where he earned the degree of Bachelor of Science in Mathematics. During the years that followed he was employed as a high school teacher in schools in Atlanta, Georgia, and Austin, Texas, where he now resides. In Summer 2001 he entered the Graduate School of Southwest Texas State University, San Marcos, Texas.

Permanent Address: 5505 Cordell Lane

Austin, Texas 78723

This thesis was typed by Steven Greenstein.