Variations on the Four-Post Tower of Hanoi Puzzle

PAUL K. STOCKMEYER
Department of Computer Science
College of William and Mary
Williamsburg, Virginia 23187-8795

0. Introduction

The famous Tower of Hanoi puzzle, invented in 1883 by Édouard Lucas (see [21]), consists of three posts and a set of n, typically 8, 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 post to post, subject to the rule that no disk can ever be placed on top of a smaller disk. It is well known that $2^n - 1$ moves are necessary and sufficient to carry out this task.

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. In this paper we survey what is known and not known about those versions of the puzzle that use four posts, rather than three.

1. The Reve's Puzzle

The first four post version was proposed by Henry Dudeney in the first chapter of his famous book, *The Canterbury Puzzles* [7]. 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 suggests that the reader try to accomplish this in the minimum number of moves, with the number of cheeses being 8, 10, and 21.

In the solution section, 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. Moreover, he sketches an answer for all cases in which the number of cheeses is a triangular number. Let $t_k = \frac{k(k+1)}{2}$ denote the k-th triangular number, and let $M(n)$ be the number of moves needed to move a tower of n cheeses. Then Dudeney claims that $M(t_k) = 2M(t_{k-1}) + 2^k - 1$, with $M(1) = 1$. This leads to $M(3) = 2 \times 1 + 2^2 - 1 = 5$, $M(6) = 2 \times 5 + 2^3 - 1 = 17$, $M(10) = 2 \times 17 + 2^4 - 1 = 49$, and so forth. No cheese-moving algorithm is given to support these numbers, and the reader is offered no help for non-triangular numbers such as 8.

This puzzle next appeared in 1939 as Problem 3918 in the *American Mathematical Monthly* [27], where it was generalized to an arbitrary number of posts. The published solutions by J. S. Frame [12] and the proposer, B. M. Stewart [28], are essentially the same. For four posts, they proposed the following algorithm scheme, based on an integer parameter i satisfying $1 \le i \le n$:

CONGRESSUS NUMERANTIUM **102** (1994), pp. 3–12

1. Recursively transport a stack consisting of the $n-i$ 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 their temporary post to the final post, again using all four posts in the process.

In addition, they proved that if n is equal to the triangular number t_k, 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. Thus they presented an explicit algorithm for this puzzle in which the number of moves used is in agreement with the partial solution of Dudeney given above.

The problems editor of the *Monthly* [8] quite properly pointed out, though, that the proof of optimality only applies to algorithms of the general scheme described above. In other words, the solvers proved that among all possible values of i, the ones specified minimize the number of moves. But they gave no justification for their assumption that an optimal algorithm must be of this form. Such a justification is still lacking today, and the Frame-Stewart solution should properly be referred to as the "presumed optimal" solution. Numerous others have rediscovered this algorithm over the years, in [1], [2], [3], [4], [5], [9], [10], [11], [19], [23], [24], [25], and [29]; many of these failed to derive the correct value for the parameter i, most mistakenly thought that they had actually proved optimality, and almost none contributed anything new to what was done by Frame and Stewart. Recently several people have presented iterative algorithms equivalent to the Frame-Stewart algorithm, in [13], [14], [18], and [20]. But the optimality of the Frame-Stewart algorithm remains a conjecture. Donald Knuth is reported to have dubbed this "Frame's conjecture" and to have written "I doubt if anyone will ever resolve the conjecture; it is truly difficult;" see [22].

Our present contribution to this puzzle is to derive a relatively simple exact closed form expression for the number $M(n)$ of moves made by the Frame-Stewart algorithm for a tower of n disks. This expression solves the recurrence relation $M(n) = 2M(n-k) + 2^k - 1$ implicit in that algorithm.

Lemma 1. *If n is one of the k numbers in the range $t_{k-1} < n \le t_k$, then*

$$M(n) - M(n-1) = 2^{k-1}.$$

This result appears in the solutions of both Frame and Stewart, and has been rediscovered by many others.

Lemma 2. *If $t_{k-1} < n \le t_k$, then*

$$M(n) = \sum_{i=1}^{k} i2^{i-1} - (t_k - n)2^{k-1}.$$

Proof: If we add up the marginal cost, in moves, of each disk from 1 to t_k, as given by Lemma 1, we obtain the given sum. As n might be less than t_k, we must then subtract the marginal cost of the extra $t_k - n$ disks.

Lemma 3. *Let $\{x\}$ be the integer nearest the real number x. Then $t_{k-1} < n \le t_k$ if and only if $k = \{\sqrt{2n}\}$.*

Proof: We have

$$\begin{aligned} k = \{\sqrt{2n}\} &\Leftrightarrow k - \tfrac{1}{2} < \sqrt{2n} < k + \tfrac{1}{2} \\ &\Leftrightarrow k^2 - k + \tfrac{1}{4} < 2n < k^2 + k + \tfrac{1}{4} \\ &\Leftrightarrow t_{k-1} + \tfrac{1}{8} < n < t_k + \tfrac{1}{8} \\ &\Leftrightarrow t_{k-1} < n \le t_k, \end{aligned}$$

where the last equivalence uses the fact that t_{k-1}, t_k, and n are all integers.

Notice this lemma implies that $\{\sqrt{2n}\}$ is an optimizing value for the parameter i in the algorithm.

Theorem 1. *Let $\theta = \{\sqrt{2n}\} - \sqrt{2n}$, so that $-\frac{1}{2} < \theta < \frac{1}{2}$. Then*

$$M(n) = 2^{\sqrt{2n}}(\alpha\sqrt{n} - \beta) + 1,$$

where $\alpha = \alpha(\theta) = 2^{\theta+1/2}(\frac{3-2\theta}{4})$ and $\beta = \beta(\theta) = 2^{\theta}(\frac{\theta^2-3\theta+4}{4})$.

Proof: Let $k = \{\sqrt{2n}\} = \sqrt{2n} + \theta$. Then using lemmas 2 and 3 and standard summation techniques we have

$$\begin{aligned} M(n) &= \sum_{i=1}^{k} i2^{i-1} - (t_k - n)2^{k-1} \\ &= \big((k-1)2^k + 1\big) - (\tfrac{k(k+1)}{2} - n)2^{k-1} \\ &= 2^k \left(\tfrac{2n-k^2+3k-4}{4}\right) + 1 \\ &= 2^{\sqrt{2n}+\theta} \left(\tfrac{2n-(\sqrt{2n}+\theta)^2+3(\sqrt{2n}+\theta)-4}{4}\right) + 1 \\ &= 2^{\sqrt{2n}+\theta} \left(\sqrt{2n}\,(\tfrac{3-2\theta}{4}) - (\tfrac{\theta^2-3\theta+4}{4})\right) + 1 \\ &= 2^{\sqrt{2n}}\big(\alpha(\theta)\sqrt{n} - \beta(\theta)\big) + 1, \end{aligned}$$

as claimed.

The functions $\alpha(\theta)$ and $\beta(\theta)$, although they look rather messy, are both in fact quite close to 1 for $-\frac{1}{2} \le \theta \le \frac{1}{2}$. The function $\alpha(\theta)$ has a minimum value of 1 at $\theta = \pm\frac{1}{2}$ and reaches a maximum of $2/(e\ln(2)) \approx 1.0615$ at $\theta = \frac{3}{2} - \frac{1}{\ln(2)} \approx 0.0573$. The function $\beta(\theta)$ is strictly decreasing from a maximum of $\frac{23}{32}\sqrt{2} \approx 1.0164$ at $\theta = -\frac{1}{2}$ to a minimum of $\frac{11}{16}\sqrt{2} \approx 0.9723$ at $\theta = \frac{1}{2}$. For n a triangular number, $n = t_k$, we have $\theta \to -\frac{1}{2}$ as $k \to \infty$. Thus $M(n) \approx 2^{\sqrt{2n}}(\sqrt{n} - 1) + 1$ is an excellent approximation for such numbers. In any case, $M(n)$ is always within 6.2% of $\sqrt{n}\,2^{\sqrt{2n}}$ for large n.

We note that $\sqrt{n}\,2^{\sqrt{2n}}$ grows strictly faster than does $\sqrt{n}\,2^{\sqrt{n}}$, so the claim of Krishnamoorthy and Biswas[15] that $M(n) = O(n^{1/2}\,2^{\sqrt{n}})$ is inconsistent with our theorem.

2. The Cyclic Puzzle

A variation of the four post puzzle was proposed in 1944 by Scorer, Grundy, and Smith [26]. In this version, the disks can only be moved along a directed cycle, from post A to post B, from B to C, from C to D, and from D to A. The authors propose the following algorithm to accomplish the task of transporting a tower of n disks 2 steps along the cycle, say from post A to post C:

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. Recursively transport the $n-1$ smallest disks from post C to post A;
4. Move the largest disk from post B to post C;
5. Recursively 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, we obtain the recurrence relation $N(n) = 3 \times N(n-1) + 2$ for $n \geq 1$, with $N(0) = 0$. The solution to this relation is easily seen to be $N(n) = 3^n - 1$.

No one seems to have noticed before that while this algorithm does indeed accomplish the specified task, it does not do so in the minimum number of moves. For 3 disks, the above algorithm makes $3^3 - 1 = 26$ moves; the reader is invited to discover any of the 48 different move sequences that will do the job in only 18 moves. Hint: it is not necessary that the top $n-1$ disks all be stacked on post C when the largest disk is first moved; they can be distributed on posts C and D in any fashion. Likewise, the only requirement that must be satisfied when the largest disk is moved from B to C is that each smaller disk reside on either post D or post A.

What can be said about this puzzle? Well, clearly the number of moves made by each disk is congruent to 2 modulo 4. Thus the total number of moves in any solution, minimal or not, will always be even. Moreover, each disk must make strictly more moves than any larger disk makes. This gives us a lower bound of $\sum_{i=1}^{n}(4(i-1)+2) = 2n^2$ for the number of moves required. For n from 1 to 7, this bound is 2, 8, 18, 32, 50, 72, and 98. A better lower bound can be obtained for $N \geq 45$, assuming Frame's conjecture, by realizing that this restricted four post puzzle can not be solved in fewer moves that the unrestricted Reve's puzzle. The upper bound of $N(n)$ provided by the algorithm is 2, 8, 26, 80, 242, 728, and 2186 respectively. An exhaustive search gives the actual minimum possible number of moves as 2, 8, 18, 36, 66, 120, and 210, respectively.

All attempts to find a compact algorithm that solves this puzzle in the minimum number of moves have so far been fruitless. There seem to be several reasons for this. First, the situation is confused by the large number of minimal move sequences. There are 640 minimal move sequences for $n = 4$, 2688 for $n = 5$, and a whopping 54,839,936 for $n = 6$. Which of these should be picked as a base to be extended into a general algorithm? There are too many to choose from.

In an attempt to limit the number of minimal sequences to be considered for generalization, we tried looking at the symmetric sequences. These are the ones in which every disk is on either post B or D at the halfway point, and the moves during the second half are the mirror images of moves during the first half, but in reverse order. Note that the original (non-minimal) algorithm generated symmetric move sequences. The number of symmetric minimal move sequences for $n = 1$, 2, 3, and 4 is 1, 2, 4, and 8, respectively. This looks promising and manageable. However, the good news stops there: none of the 2,688 minimal move sequences for $n = 5$ are symmetric, and none of the 54,839,936 minimal move sequences for $n = 6$ are symmetric! In fact, there seems to be no apparent rhyme or reason, no predictable structure, in these minimal sequences.

A semi-brute-force approach was also tried. There is an algorithm, which might be called the bottom-driven algorithm, that goes something like this:

for each disk d from n to 1 do
    while disk d isn't on the desired post,
    arrange for it to be movable and move it.

In describing this algorithm we follow the normal convention that the disks are numbered in order from 1 to n with 1 the smallest and n the largest. The "arrange for it to be movable" part can easily be accomplished by recursion, moving each disk from $n-1$ to 1 as necessary to the first available post that doesn't conflict with the proposed move of disk d. This algorithm was first proposed in the context of moving from a random distribution of n disks on 3 posts to the goal configuration of all disks on a designated post. It has been proved that this algorithm always generates a minimal length move sequence for the puzzle that motivated it, provided, of course, that the initial configuration has the disks on each post arranged in the standard order of size.

Once again, things go well for $n = 1$, 2, 3, and 4. The bottom-driven algorithm does indeed produce one of the minimal sequences of moves in each case. But for $n = 5$, this algorithm makes 70 moves, rather than the minimal 66.

Yet another attempt might be called "ignore the smallest." In this approach the n disk puzzle is solved by applying the $n-1$ disk algorithm to the largest $n-1$ disks, pushing the smallest disk along the cycle whenever it is in the way of the next move to be made. But for the 4-post cyclic puzzle this turns out to be equivalent to the bottom-driven algorithm.

In short, we have not been able to find any algorithm for this puzzle that is significantly better than a brute force breadth-first search of the state graph of the puzzle.

3. The Four-in-a-Row Puzzle

The same paper [26] that posed the 4-post cyclic puzzle also mentioned a restricted three post puzzle. Imagine three posts A, B, and C in a row. Moves are allowed between posts A and B, and between B and C, in either direction, but not between posts A and C. The challenge is to transport a tower of n disks from post A to post C in the minimum number of moves.

This turns out to be a rather trivial puzzle. The state graph, with the possible states of the puzzle as vertices and the allowable state transformations as edges, is simply one long path containing all 3^n states, with the start state and the goal state at opposite ends. Moving the tower thus takes $3^n - 1$ moves, and it is hard to go wrong. But it does suggest another four post variation, in which the allowable moves are back and forth between posts A and B, between B and C, and between C and D. The object is to transport a tower along the row from post A to post D.

This four-in-a-row puzzle does not appear to have been studied elsewhere in the literature, and we have done only a very preliminary investigation. We establish an upper bound on the number of moves required for this puzzle by presenting an effective but non-optimal algorithm for its solution:

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 post D to post B using the three-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 three-in-a-row algorithm, again ignoring post A.

Using $R(n)$ to denote the number of moves made by this algorithm for the row puzzle with n disks, we have the recurrence relation $R(n) = R(n-1) + 2 + (3^{n-1} - 1) + 1 + (3^{n-1} - 1) = R(n-1) + 2 \times s^{n-1} + 1$ for $n \geq 1$, with $R(0) = 0$. The solution to this relation is $R(n) = 3^n + n - 1$, implying that the minimum number of moves required for this puzzle is no worse than $O(3^n)$. Again, a lower bound is supplied by the solution to the Reve's puzzle.

Further evidence suggests that this puzzle is every bit as hard to handle as the cyclic puzzle, and has much the same feel. Clearly each disk must make an odd number of moves, so the total number of moves made has the same parity as n. For n from 1 to 6, the number of moves required is 3, 10, 19, 34, 57, and 88, but already with 6 disks we have 2,097,152 different minimal move sequences. There is no such thing as a symmetric move sequence here. It is not clear how to write a bottom-driven algorithm for this puzzle, and there is no guarantee that such an algorithm would be minimal.

As with the cyclic puzzle, we have not been able to find an algorithm that is significantly better than a breadth-first search of the state graph.

4. The Star Puzzle

Clearly, the arc set of every strongly connected digraph on four vertices could be considered a representation of the allowable moves of some four post tower puzzle. Some non-strong digraphs will work as well; see [16] and [17]. So far we have considered the complete digraph on four vertices, the 4-cycle, and the path; there are over 80 more to go. But with the great proliferation of new tower puzzles posed during the past 20 years, perhaps the author of a new variation should be called upon to justify inflicting it on a world already swimming in similar puzzles. Our justifications for the new puzzle we introduce in this section are that 1) it is at least as entertaining and attractive a puzzle as any other considered here; 2) there is an algorithm, which we conjecture is optimal, that possess a certain amount of elegance and charm; and 3) it is similar enough to the (unrestricted) Reve's puzzle that it may eventually shed light on how to show that the presumed optimal solution there is in fact truly minimal.

Our new puzzle consists of three posts, labeled A, B, and C, arranged in an equilateral triangle, and a fourth post, labeled O, in the middle. Every disk move must be either to or from post O; direct moves between any two of posts A, B, and C are prohibited. Thus the allowable move graph is a star. The task is to transport a tower of n disks from post A to, say, post C.

Of course we could just carry out a minimal solution to the original three post tower puzzle, with each disk move taking a slight detour through post O. This would take $2 \times (2^n - 1) = 2^{n+1} - 2$ moves. But in fact we can do much better. Consider the following algorithm scheme which, like the scheme for the Reve's puzzle, has an integer parameter i in the range $1 \leq i \leq n$:

1. Recursively transport a stack consisting of the $n - i$ smallest disks from post A to post B, using all four posts in the process;
2. Transport the stack consisting of the i largest disks from post A to post C, using the three-in-a-row algorithm from the previous section and ignoring post B;
3. Recursively transport the smallest $n - i$ disks from post B to post C, again using all four posts in the process.

If we use $S(n)$ to denote the minimum number of moves made by this algorithm scheme for the star puzzle with n disks, we have

$$S(n) = \min_{1 \leq i \leq n} \left(2S(n-i) + 3^i - 1\right)$$

for $n \geq 1$, with $S(n) = 0$. The naive algorithm, mimicking the standard three post tower puzzle, corresponds to picking $i = 1$ always in the above scheme. This is optimal for $n = 1$ and 2, but the minimizing value for i is easily seen to be 2 when $n = 3$, 4, 5, and 6, and to be 3 or move for $n \geq 7$. Our next theorem tells the whole story.

Theorem 2. *Let $\{a_m\}_{m=1}^{\infty}$ be the sequence $1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, \ldots$ of integers of the form $2^j 3^k$, with $j \geq 0$ and $k \geq 0$, in increasing order. Then*

$$i = \lfloor \log_3(a_n) \rfloor + 1$$

is the unique minimizing value for i in the algorithm scheme for the star puzzle. In the resulting algorithm, there is a disk making exactly $2a_m$ moves for each m from 1 to n, resulting in

$$S(n) = 2 \times \sum_{m=1}^{n} a_m.$$

Proof: The proof is by induction on n. The theorem is easily seen to be true for small values; we assume it is true for all tower sizes less than some arbitrary n. The optimal solution for the three-in-a-row puzzle tells us that, counting up from the bottom, the largest i disks make 2, 6, 18, …, and $2 \times 3^{i-1}$ moves. The remaining $n - i$ disks are involved in two recursive calls, and so by the induction hypothesis, these disks make $4a_1$, $4a_2$, $4a_3, \ldots,$ and $4a_{n-i}$ moves. Now these two sequences are disjoint, and if extended infinitely, they would exactly partition the sequence

$\{2 \times a_m\}$. In order for the sums of these two sequences to be minimal, there must be no gaps in their union; that is, together they must exactly cover the sequence $2a_1, 2a_2, 2a_3, \ldots, 2a_n$. Concentrating on the last element of the first sequence, we see that this will happen if and only if $2 \times 3^{i-1} \leq 2a_n < 2 \times 3^i$. Taking logarithms, we have $i - 1 \leq \log_3(a_n) < i$, or $i = \lfloor \log_3(a_n) \rfloor + 1$ as claimed.

It is important to make it clear what we have proved. Theorem 2 identifies and analyzes the optimal algorithm within the given algorithm scheme. But as with the Reve's puzzle, there is no proof that an optimal algorithm for the star puzzle must be of this form. We strongly suspect, though, that this algorithm is indeed optimal, and we will perhaps somewhat impertinently call it the "presumed optimal" algorithm for this puzzle.

5. How hard are these Puzzles?

Are these puzzles inherently difficult? During the years 1984-1986 a rather childish debate raged in *The Computer Journal* over the question "is the tower of Hanoi a trivial problem?" A more serious attempt to answer this question, based on complexity theory, was presented by Cull and Gerety [6] at about the same time. Drawing somewhat from their ideas, we will consider two problems that can be posed about any tower puzzle.

The *listing problem* is to produce a list showing the starting post and the destination post of each move in a minimal solution, given the number n of disks. We will call the listing problem *easy* for a tower puzzle if there exists an algorithm for the listing problem with running time no worse than a polynomial function of the number of moves. Otherwise the listing problem is *hard* for that puzzle. The *size problem* accepts two integers n and k as input, and outputs *yes* or *no* depending on whether or not the n-disk instance of the tower puzzle in question can be solved in at most k moves. This problem is *easy* if there exists an algorithm that answers it with running time no worse than a polynomial function of input size, and is *hard* otherwise. Cull and Gerety argue that both of these problems are easy for the standard three post tower puzzle, and conjecture that all properly posed tower problems are easy for the standard puzzle.

These problems have not been studied for the four post puzzles, but some reasonable guesses can be made. If the presumed optimal algorithms for the Reve's puzzle is in fact optimal, then it seems quite likely that both problems are easy for this puzzle. Almost any line in the proof of Theorem 1 can be used to solve the size problem efficiently, and we have an explicit algorithm for producing a listing of moves. The situation is less clear for the star problem. We so far have no closed form expression for the number of moves needed, and no fast means of computing the optimizing value of the parameter i in the algorithm, even assuming that the proposed optimal solution is truly optimal. Nevertheless, we suspect that the two problems are both easy for this puzzle as well.

The situation is quite different for the 4-post cyclic puzzle and the four- in-a-row puzzle. Here we have no explicit algorithm, with or without some parameter, for solving the puzzle in the minimum number of moves. We don't know whether the required number of moves grows exponentially with n, or only sub-exponentially, as with the Reve's puzzle. These two are likely candidates for puzzles with associated problems that are really hard. A breadth-first search of the state graph for either of these puzzles might have to visit nearly all of the 4^n vertices before discovering a minimal move sequence of length only $O(3^n)$. To view the situation another way, a personal computer can determine that $M(100) = 172,033$ and $S(100) = 2,792,570$ in the blink of an eye. But it is quite possible that we may never know exactly how many moves are required for the four post cyclic puzzle or the four-in-a-row puzzle for $n = 100$ disks.

The one conclusion that we draw from this discussion is that the problem of determining which of these puzzles have easy problems associated with them, and which have hard, will likely be very difficult to solve.

References.

[1.] Boardman, J. T., C. Garrett, and G. C. A. Robson, A Recursive Algorithm for the Optimal Solution of a Complex Allocation Problem using a Dynamic Programming formulation, *The Computer Journal 29* (1986), 182–186.

[2.] Brousseau, Brother Alfred, Tower of Hanoi with More Pegs, *J. Recreational Mathematics 81* (1975-76), 169–176.

[3.] Celko, Joe, Mutants of Hanoi. *ABACUS 1* (1984), 54–57.

[4.] Chu, I-Ping, and Richard Johnsonbaugh, The Four-Peg Tower of Hanoi Puzzle, *SIGCSE Bulletin 23* (1991), 2–4.

[5.] Cull, Paul, and E. F. Ecklund, Jr., On the Towers of Hanoi and Generalized Towers of Hanoi Problems, *Congressus Numerantium 35* (1982), 229–238.

[6.] Cull, Paul, and Colin Gerety, Is Towers of Hanoi Really Hard?, *Congressus Numerantium 47* (1985), 237–242.

[7.] Dudeney, Henry Ernest, *The Canterbury Puzzles (and other curious problems).* Thomas Nelson & Sons, London, 1907.

[8.] Dunkel, O., Editorial Note concerning Advanced Problem 3918, *American Mathematical Monthly 48* (1941), 219.

[9.] Er, M. C., An Algorithmic Solution to the Multi-tower Hanoi Problem, *J. Inform. Optim. Sciences 8* (1987), 91–100.

[10.] Er, M. C., An Optimal Algorithm for Reve's Puzzle, *Information Sciences 45* (1988), 39–49.

[11.] Er, M. C., A Note on the Optimality of a Reve Algorithm, *The Computer Journal 34* (1991), 513.

[12.] Frame, J. S., Solution to Advanced Problem 3918, *American Mathematical Monthly 48* (1941), 216–217.

[13.] Gedeon, T. D., The Reve's Puzzle: An iterative solution produced by transformation, *The Computer Journal 35* (1992), 186–187.

[14.] Hinz, A. M., An Iterative Algorithm for the Tower of Hanoi with Four Pegs, *Computing 42* (1989), 133–140.

[15.] Krishnamoorthy, M. S., and S. Biswas, The Generalized Towers of Hanoi, *SIGACT News 10* (Winter 1979), 49.

[16.] Leiss, Ernst L., Solving the "Towers of Hanoi" on Graphs, *J. Combinatorics, Information & System Sciences 8* (1983), 81-89.

[17.] Leiss, Ernst L., On Restricted Hanoi Problems, *J. Combinatorics, Information & System Sciences 8* (1983), 277-285.

[18.] Liefvoort, A. van de, An Iterative Algorithm for the Reve's Puzzle, *The Computer Journal 35* (1992), 91–92.

[19.] Lu, Xue-Miao, Towers of Hanoi Problem with Arbitrary $k \geq 3$ Pegs, *Intern. J. Computer Math. 24* (1988), 39–54.

[20.] Lu, Xue-Miao, An Iterative Solution for the 4-peg Towers of Hanoi, *The Computer Journal 32* (1989), 187–189

[21.] Lucas, Édouard, *Récréations Mathématiques, vol. III.* Gauthier-Villars, Paris, 1893, pp. 55–59. Reprinted several times by Albert Blanchard, Paris.

[22.] Lunnon, W. F., The Reve's Puzzle, *The Computer Journal 29* (1986), 478.

[23.] Rohl, J. S., and T. D. Gedeon, Four-Tower Hanoi and Beyond, *Proc. 6-th Austral. Comput. Sci. Conf.* Sidney, 1983, 156–162.

[24.] Rohl, J. S., and T. D. Gedeon, The Reve's Puzzle, *The Computer Journal 29* (1986), 187–188. Corregendum, 31 (1988), 190.

[25.] Roth, Ted, The Tower of Brahma Revisited, *J. Recreational Mathematics 7* (1974), 116–119.

[26.] Scorer, R. S., P. M. Grundy, and C. A. B. Smith, Some Binary Games, *The Mathematical Gazette 280* (1944), 96–103.

[27.] Stewart, B. M., Advanced Problem 3918, *American Mathematical Monthly 46* (1939), 363.

[28.] Stewart, B. M., Solution to Advanced Problem 3918, *American Mathematical Monthly 48* (1941), 217–219.

[29.] Wood, Derick, The Towers of Brahma and Hanoi revisited, *J. Recreational Mathematics 14* (1981-82), 17–24.