Simple Variations on the Tower of Hanoi to Guide the Study of Recurrences and Proofs by Induction

Saad Mneimneh
Department of Computer Science
Hunter College, The City University of New York
695 Park Avenue, New York, NY 10065 USA
saad@hunter.cuny.edu

***Abstract*— *Background*: The Tower of Hanoi problem was formulated in 1883 by mathematician Edouard Lucas. For over a century, this problem has become familiar to many of us in disciplines such as computer programming, data structures, algorithms, and discrete mathematics. Several variations to Lucas' original problem exist today, and interestingly some remain unsolved and continue to ignite research questions.**
Research Question: **Can this richness of the Tower of Hanoi be explored beyond the classical setting to create opportunities for learning about recurrences and proofs by induction?**
Contribution: **We describe several simple variations on the Tower of Hanoi that can guide the study and illuminate/clarify the pitfalls of recurrences and proofs by induction, both of which are an essential component of any typical introduction to discrete mathematics and/or algorithms.**
Methodology and Findings: **Simple variations on the Tower of Hanoi lead to different interesting recurrences, which in turn are associated with exemplary proofs by induction.**

***Index Terms*—Tower of Hanoi, Recurrences, Proofs by Induction.**

I. Introduction

THE Tower of Hanoi problem, formulated in 1883 by French mathematician Edouard Lucas [8], consists of three vertical pegs, labeled x, y, and z, and n disks of different sizes, each with a hole in the center that allows the disk to go through pegs. The disks are numbered $1, \dots, n$ from smallest to largest. Initially, all n disks are stacked on one peg as shown in Figure 1, with disk n at the bottom and disk 1 at the top.

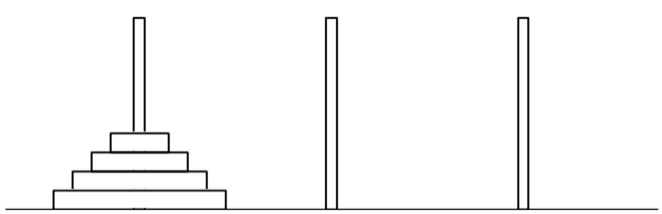

Fig. 1. Lucas' Tower of Hanoi for $n = 4$.

The goal is to transfer the entire stack of disks to another peg by repeatedly moving one disk at a time from a source peg to a destination peg, and without ever placing a disk on top of a smaller one. The physics of the problem dictate that a disk can be moved only if it sits on the top of its stack. The third peg is used as a temporary place holder for disks while they move throughput this transfer.

The classical solution for the Tower of Hanoi is recursive in nature and proceeds to first transfer the top $n-1$ disks from peg x to peg y via peg z, then move disk n from peg x to peg z, and finally transfer disks $1, \dots, n-1$ from peg y to peg z via peg x. Here's the (pretty standard) algorithm:

```
Hanoi(n, x, y, z)
  if n > 0
     then Hanoi(n − 1, x, z, y)
          Move(1, x, z)
          Hanoi(n − 1, y, x, z)
```

In general, we will have a procedure *Transfer*$(n, from, via, to)$ to (recursively) transfer a stack of height n from peg $from$ to peg to via peg via, which will be named according to the problem variation (it's Hanoi above), and a procedure Move$(k, from, to)$ to move the top k (1 in Hanoi) from peg $from$ to peg to (in one move). We will also use Exchange(i) to exchange disk i and disk 1 (see Section VI).

An analysis of the above strategy is typically presented by letting a_n be the total number of moves, and establishing the recurrence $a_n = a_{n-1} + 1 + a_{n-1} = 2a_{n-1} + 1$, where $a_1 = 1$. The recurrence is iterated for several values of n to discover a pattern that suggests the closed form solution $a_n = 2^n - 1$. This latter expression for a_n is proved by induction using the base case above and the recurrence itself to carry out the inductive step of the proof.

Perhaps a very intriguing thought is that we need about 585 billion years to transfer a stack of 64 disks [1] if every move requires one second! This realization is often an aha moment on a first encounter, and illustrates the impact of exponential growth on complexity. Though not immediately obvious, other interesting facts can also be observed (and proved):

- **Fact 1**. When $n = 0$ (empty stack of disks), $a_0 = 2^0 - 1 = 0$ is consistent in that we need zero moves to transfer all of the disks (none in this case).
- **Fact 2**. The number of moves $a_n = 2^n - 1$ is optimal, i.e. $2^n - 1$ represents the smallest possible number of moves to solve the problem (and the optimal solution is unique).
- **Fact 3**. Disk i must make at least 2^{n-i} moves (exactly 2^{n-1} in the optimal solution). Observe that $\sum_{i=1}^{n} 2^{n-i} =$

[1] The case of $n = 64$ is related to a myth about 64 golden disks and the end of the world [5].

$2^n - 1$ for $n \geq 0$ (again touching on the notion of the empty sum when $n = 0$).
- **Fact 4**. If we define a *repeated* move as the act of moving the same disk from a given peg to another given peg, then every solution must have a repeated move when $n \geq 4$ (pigeonhole principle applied to the moves of disk 1).
- **Fact 5**. There is a (non-optimal) solution for $n = 3$ with eight moves none of which are repeated moves.

The above facts highlight some richness of the problem as they touch on several aspects of mathematical and algorithmic flavor which, when pointed out, can be very insightful. For instance, Fact 1 is a good reminder of whether to associate a 0 or a 1 when dealing with an empty instance of a given problem (empty sum vs. empty product). Fact 2 directs our attention to what is optimal (not just feasible), and Fact 3 is a more profound way to address that optimality. Fact 4 is a nice application of the pigeonhole principle that requires knowledge of Fact 3; a weaker version can be proved, namely when $n \geq 5$ instead of $n \geq 4$, if we only rely on Fact 2. Finally, Fact 5 raises the question of how things may be done differently when we seek non-typical answers. Along those lines, several variations for the Tower of Hanoi already exist, which include a combination of restricted moves, colored disks, multiple stacks, and multiple pegs. For instance, we refer the reader to [11], [12], [13], [7], [2], [4], [1], [5] for some literature and examples.

Here we explore several variations while sticking to the one stack of disks and three pegs. Our goal is not to extend the research on the Tower of Hanoi problem but rather provide simple, and yet interesting, variants of it to guide (and enrich) the study of recurrences and proofs by induction in introductory discrete mathematics. Therefore, we assume basic familiarity with mathematical induction and solving linear recurrences of the form

$$a_n = p_1 a_{n-1} + p_2 a_{n-2} + \ldots + p_k a_{n-k} + f(n)$$

Several techniques for solving recurrences can be used, such as making a guess and proving it by induction (e.g. $a_n = 2a_{n-1} + 1$), summing up $\sum_{i=0}^{n} a_i$ to cancel out factors and express a_n in terms of a_1 or a_0 (e.g. $a_n = a_{n-1} + 2^n$), applying a transformation to achieve the desired form (e.g. $a_n = 2a_{n/2} + n - 1$ and take $n = 2^k$, or $a_n = 3a_{n-1}^2$ and let $b_n = \log a_n$), generating functions (of the form $g(x) = a_0 + a_1 x + a_2 x^2 + \ldots = \sum_{n=0}^{\infty} a_n x^n$), etc... [9].

In particular, the method of using the characteristic equation $x^k = \sum_{i=1}^{k} p_i x^{k-i}$ when $f(n) = 0$ (homogeneous recurrence) is systematic and suitable for an introductory level. For example, when $a_n = p_1 a_{n-1} + p_2 a_{n-2}$ (a second order homogeneous linear recurrence), and r_1 and r_2 are the roots of $x^2 = p_1 x + p_2$, then $a_n = c_1 r_1^n + c_2 r_2^n$ if $r_1 \neq r_2$, and $a_n = c_1 r^n + c_2 n r^n$ if $r_1 = r_2 = r$. We can solve for the constants c_1 and c_2 by making a_n satisfy the boundary conditions; for instance, a_1 and a_2 for $n = 1$ and $n = 2$, respectively. This technique can be generalized to homogeneous linear recurrences of higher orders.

When $f(n) \neq 0$ (non-homogeneous recurrence), we try to find an equivalent homogeneous recurrence, by annihilating the term $f(n)$. For instance, the recurrence for the Lucas' Tower of Hanoi problem satisfies $a_n = p_1 a_{n-1} + f(n)$, where $p_1 = 2$ and $f(n) = 1$, so it's non-homogeneous, but subtracting a_{n-1} from a_n gives $a_n - a_{n-1} = 2a_{n-1} - 2a_{n-2}$, which yields the homogeneous recurrence $a_n = 3a_{n-1} - 2a_{n-2}$.

II. DOUBLE DECKER

In this variation, called Double Decker, we duplicate every disk to create a stack of $2n$ disks with two of each size as shown in Figure 2. For convenience of notation, we will consider (only for this variant) that a stack of height n has $2n$ disks.

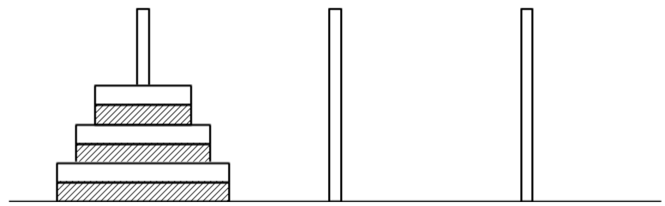

Fig. 2. Double Decker for $n = 3$, suggested in [3].

A trivial solution to Double Decker is to simply treat it as a standard instance of the Tower of Hanoi with $2n$ disks and, thus, will need the usual $2^{2n} - 1 = 4^n - 1$ moves. This trivial solution, however, does not benefit from equal size disks. For instance, if we do not require that disks of the same size must preserve their original order, then a better solution for Double Decker is to emulate the standard Tower of Hanoi solution by duplicate moves, to give $a_0 = 0$, $a_1 = 2$, $a_2 = 6$, ... The algorithm is shown below.

```
DoubleDecker(n, x, y, z)
  if n > 0
     then DoubleDecker(n-1, x, z, y)
          Move(1, x, z)
          Move(1, x, z)
          DoubleDecker(n-1, y, x, z)
```

The Double Decker recurrence is $a_n = 2a_{n-1} + 2$ and, since we expect that the solution now requires twice the number of original moves, we can use that recurrence to show by induction that $a_n = 2(2^n - 1) \ll 4^n - 1$. The inductive step for $n = m > 0$ will be as follows:

$$a_m = 2a_{m-1} + 2 = 2 \cdot 2(2^{m-1} - 1) + 2$$
$$= 2 \cdot 2^m - 4 + 2 = 2 \cdot 2^m - 2 = 2(2^m - 1)$$

with $a_0 = 0$ as a base case (examples of making a careful choice of base case(s) will follow throughout the exposition).

Alternatively, we can solve the recurrence itself, by first changing it into a homogeneous recurrence using the technique outlined in the previous section, to obtain $a_n = 3a_{n-1} - 2a_{n-2}$ with the characteristic equation $x^2 = 3x - 2$, and the roots $r_1 = 1$ and $r_2 = 2$. So we write $a_n = c_1 1^n + c_2 2^n$ and solve for c_1 and c_2 using a_n for two values of n; for instance,

$$a_0 = c_1 1^0 + c_2 2^0 = c_1 + c_2 = 0$$
$$a_1 = c_1 1^1 + c_2 2^1 = c_1 + 2c_2 = 2$$

which will result in $c_1 = -2$ and $c_2 = 2$.

An interesting subtlety about Double Decker is to observe that, although we did not require to preserve the original order of disks (so long no disk is placed on a smaller one), the above solution only switches the bottom two disks (disks $2n-1$ and $2n$). This can be verified by Fact 3: since disk i of the original Tower of Hanoi must make 2^{n-i} moves, and that's an even number when $i < n$, disks $2i-1$ and $2i$ in Double Decker must do the same, and hence will preserve their order. However, disks $2n-1$ and $2n$ in Double Decker will each make an odd number of moves (namely just $2^{n-n} = 1$ move), and hence will switch.

Therefore, to make Double Decker preserve the original order of all disks, we can perform the algorithm twice, which will guarantee that every disk will make an even number of moves, at the cost of doubling the number of moves.

```
DoubleDeckerTwice(n, x, y, z)
  if n > 0
     then DoubleDecker(n, x, z, y)
          DoubleDecker(n, y, x, z)
```

The total number of moves for the above algorithm is therefore twice $2(2^n - 1)$, which is $4(2^n - 1)$. But can we do better? One idea is to avoid fixing the order of the last two disks by forcing the correct order in the first place. Here's a first bad attempt.

```
DoubleDeckerBad(n, x, y, z)
  if n > 0
     then DoubleDeckerBad(n − 1, x, y, z)
          Move(1, x, y)
          DoubleDeckerBad(n − 1, z, x, y)
          Move(1, x, z)
          DoubleDeckerBad(n − 1, y, z, x)
          Move(1, y, z)
          DoubleDeckerBad(n − 1, x, y, z)
```

The DoubleDeckerBad algorithm is a simple disguise of the standard Tower of Hanoi algorithm for $2n$ disks, which was presented as algorithm Hanoi in Section I. Observe that in order to transfer $2n$ disks from peg x to peg z, we first transfer $2n-1$ disks from peg x to peg y (recursively in the first three lines following `if n > 0`), then move the top disk from peg x to peg z (in the subsequent fourth line), and finally transfer $2n-1$ disks from peg y to peg z (recursively in the last three lines). This is nothing but the standard sequence of moves for the $2n$-disk Tower of Hanoi.

In fact, it is not hard to verify that the recurrence $b_n = 4b_{n-1} + 3$ of DoubleDeckerBad has the solution $b_n = 4^n - 1$ (same as Tower of Hanoi for $2n$ disks). However, an interesting take on this is to consider the following two recurrences:

$$\begin{array}{ll} a_n = 2a_{n-1} + 1 & \text{Hanoi} \\ b_n = 4b_{n-1} + 3 & \text{DoubleDeckerBad} \end{array}$$

and show by induction that $b_n = a_{2n}$. For instance, an inductive step for $n = m > n_0$ will be

$$b_m = 4b_{m-1} + 3 = 4a_{2m-2} + 3$$

$$= 2(2a_{2m-2} + 1) + 1 = 2a_{2m-1} + 1 = a_{2m}$$

Although only one inductive step is involved (b_m and b_{m-1}), $a_n = 2a_{n-1} + 1$ was iterated twice "backwards" (a_{2m-2}, a_{2m-1}, and a_{2m}), which on the one hand is not how one would typically proceed from $a_n = 2a_{n-1} + 1$ to establish some truth about a_n, and on the other hand raises the question of whether multiple base cases are needed (what should the value of n_0 be?). The number of base cases is often a subtle detail about proofs by induction, and without it, there will be a lack of insight into the method of mathematical induction itself. Before we address this aspect of the proof, let us establish few base cases to verify their truth: $b_0 = a_{2\cdot 0} = a_0 = 0$, $b_1 = a_{2\cdot 1} = a_2 = 3$, $b_2 = a_{2\cdot 2} = a_4 = 15$, ... Typically, a person who is attempting this proof by induction will be easily inclined to verify a bunch of base cases, as this feels somewhat *safe* for providing enough evidence that the property $b_n = a_{2n}$ holds. In principle, however, one should have a systematic approach. A careful examination of the inductive step above will reveal that it works when $m - 1 \geq 0$ and $2m - 2 \geq 0$ (otherwise, b_{m-1} and a_{2m-2} are not defined). Therefore, we need $m > 0$, so $n_0 = 0$ is good enough, and we only need to verify that $b_0 = a_0$.

So far, $4(2^n - 1)$ is our smallest number of moves for solving the Double Decker Tower of Hanoi. As it turns out, we can save one more move (and we later prove optimality)! This can be done by adjusting the previous attempt not to recursively handle the two bottom disks (but only disks $2n-1$ and $2n$):

```
DoubleDeckerBest(n, x, y, z)
  if n > 0
     then DoubleDecker(n − 1, x, y, z)
          Move(1, x, y)
          DoubleDecker(n − 1, z, x, y)
          Move(1, x, z)
          DoubleDecker(n − 1, y, z, x)
          Move(1, y, z)
          DoubleDecker(n − 1, x, y, z)
```

Switching the order of equal size disks is not an issue now since DoubleDecker is called an even number of times (namely four times). The total number of moves is given by $a_n = 4 \cdot 2(2^{n-1} - 1) + 3 = 4(2^n - 1) - 1$ when $n > 0$, and $a_0 = 0$.

To prove the above is optimal, we observe that the other possible strategy is to move the largest disk to its destination from the intermediate peg (instead of the source peg).

```
DoubleDeckerAltBest(n, x, y, z)
  if n > 1
     then DoubleDecker(n − 1, x, y, z)
          Move(1, x, y)
          Move(1, x, y)
          DoubleDecker(n − 1, z, y, x)
          Move(1, y, z)
          Move(1, y, z)
          DoubleDeckerAltBest(n − 1, x, y, z)
     else Hanoi(2n, x, y, z)
```

The above algorithm generates the recurrence $a_n = 2(2^{n-1} - 1) + 2 + 2(2^{n-1} - 1) + 2 + a_{n-1} = a_{n-1} + 2^{n+1}$ when $n > 1$, which can be shown by induction to have the solution $a_n = 4(2^n - 1) - 1$, thus proving that this is optimal.

The Double Decker can be easily generalized to a k-Decker ($k > 1$) with $a_{n>0} = 2k(2^n - 1) - 1$ moves. A further generalization of k-Decker in which there are k_i disks of size i is also suggested in [3]. It is not hard to show that, based on Fact 3, this generalization requires $2(\sum_{i=1}^{n} k_i 2^{n-i}) - 1$ moves if $k_n > 1$, and $\sum_{i=1}^{n} k_i 2^{n-i}$ if $k_n = 1$, which is equal to $2k(2^n - 1) - 1$ when $k_1 = k_2 = \ldots = k_n = k > 1$, and $2^n - 1$ if $k = 1$.

III. Move One Get Some Free

In this variation, we can move the top $k \in \mathbb{N}$ or fewer disks from a given peg to another simultaneously, and still consider this to be one move. Hence the name Move One Get Some ($k-1$) Free. It is not hard to see that the optimal number of moves can be achieved by (when $n > 0$)

$$a_n = \min_{0<i\leq\min\{k,n\}} 2a_{n-i} + 1$$

$$= 2a_{n-\min\{k,n\}} + 1 = 2a_{\max\{n-k,0\}} + 1$$

since a_n must be non-increasing in n and, therefore, it is better to moves simultaneously as many disks as possible when moving the largest to its destination. The above recurrence is simply $a_n = 2a_{n-k} + 1$ when $n \geq k$. As such, we can show that $a_n = 2^{\lceil n/k \rceil} - 1$, which amounts to breaking the original stack of disks into $\lceil n/k \rceil$ virtual disks, each consists of k or fewer disks. The algorithm for this variation is shown below:

```
MoveOneGetSomeFree(n, x, y, z)
  if n > 0
     then MoveOneGetSomeFree(n − k, x, z, y)
          Move(min{k, n}, x, z)
          MoveOneGetSomeFree(n − k, y, x, z)
```

The proof that $a_n = 2^{\lceil n/k \rceil} - 1$ is by (strong) induction for $n = m > n_0$:

$$a_m = 2a_{m-k} + 1 = 2(2^{\lceil (m-k)/k \rceil} - 1) + 1$$

$$= 2 \cdot 2^{\lceil m/k - 1 \rceil} - 1 = 2 \cdot 2^{\lceil m/k \rceil - 1} - 1 = 2^{\lceil m/k \rceil} - 1$$

Following the same line of thought from the previous section about the choice of base cases, we must ensure that we verify enough, but not too many. The inductive step requires that a_{m-k} be defined and thus $m \geq k$. So $n_0 = k - 1$, which means that we must verify all bases cases for $n = 0, \ldots, k-1$.

$$a_0 = 2^{\lceil 0/k \rceil} - 1 = 1 - 1 = 0$$

$$a_n = 2^{\lceil n/k \rceil} - 1 = 2 - 1 = 1, \quad n = 1, \ldots, k-1$$

Therefore, the Move One Get Some ($k-1$) Free generalizes the standard Tower of Hanoi (which becomes the special case when $k = 1$). Interestingly, we could also study this general variation of the Tower of Hanoi by solving the recurrence $a_n = 2a_{n-k} + 1$ itself, using the method of characteristic equations. First, we transform the recurrence into a homogeneous one, by subtracting (as outline in Section 1) $a_n - a_{n-1} = 2a_{n-k} - 2a_{n-k-1}$, which yields:

$$a_n = a_{n-1} + 2a_{n-k} - 2a_{n-k-1}$$

and the characteristic equation:

$$x^{k+1} = x^k + 2x - 2$$

By observing that $r_0 = 1$ is a root, we can express the characteristic equation as follows:

$$(x-1)(x^k - 2) = 0$$

and thus the $k+1$ (distinct) roots are $r_0 = 1$, and $r_{s+1} = \sqrt[k]{2} e^{i2\pi s/k}$ for $0 \leq s < k$ (the k^{th} roots of 2), where $e^{i\theta} = \cos\theta + i \sin\theta$. An example when $k = 5$ is shown in Figure 3.

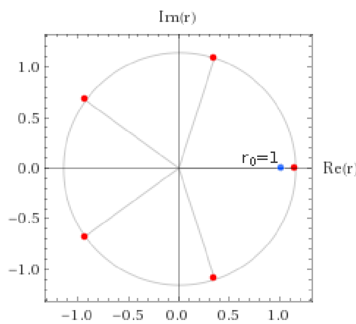

Fig. 3. The k^{th} roots of 2 when $k = 5$: $r_1 = \sqrt[5]{2}, r_2, \ldots, r_5$; and $r_0 = 1$. Generated in part by WolframAlpha at https://www.wolframalpha.com [14].

Using the above information about the roots for a given k, one can construct several interesting proofs by induction (possibly involving the complex numbers). For instance, when $k = 2$ (Move One Get One Free), we have $r_0 = 1$, $r_1 = \sqrt{2}$, and $r_2 = -\sqrt{2}$. Given the form $a_n = c_1 + c_2\sqrt{2}^n + c_3(-\sqrt{2})^n$ with $a_0 = 0$, $a_1 = 1$, and $a_2 = 1$, we obtain

$$a_0 = c_1 + c_2 + c_3 = 0$$

$$a_1 = c_1 + \sqrt{2}c_2 - \sqrt{2}c_3 = 1$$

$$a_2 = c_1 + 2c_2 + 2c_3 = 1$$

and $c_1 = -1$, $c_2 = (1+\sqrt{2})/2$, $c_3 = (1-\sqrt{2})/2$, and

$$a_n = -1 + \frac{1+\sqrt{2}}{2}\sqrt{2}^n + \frac{1-\sqrt{2}}{2}(-\sqrt{2})^n$$

Therefore, one could try to prove by induction the following for $n \geq 0$:

$$2^{\lceil n/2 \rceil} = \frac{1+\sqrt{2}}{2}\sqrt{2}^n + \frac{1-\sqrt{2}}{2}(-\sqrt{2})^n$$

which provides an interesting and not so trivial interplay of the patterns $2^{\lceil\ \rceil}$ and $\sqrt{2}$, but rather intuitive because $2^{n/2} = \sqrt{2}^n$. The proof (by strong induction) and the careful choice of base case(s) follow, given $n = m > n_0$:

$$2^{\lceil m/2 \rceil} = 2^{\lceil (m-2)/2+1 \rceil} = 2 \cdot 2^{\lceil (m-2)/2 \rceil}$$

$$= 2\Big[\frac{1+\sqrt{2}}{2}\sqrt{2}^{m-2} + \frac{1-\sqrt{2}}{2}(-\sqrt{2})^{m-2}\Big]$$

$$= 2\Big[\frac{1+\sqrt{2}}{4}\sqrt{2}^{m} + \frac{1-\sqrt{2}}{2(-\sqrt{2})^2}(-\sqrt{2})^{m}\Big]$$

$$= \frac{1+\sqrt{2}}{2}\sqrt{2}^{m} + \frac{1-\sqrt{2}}{2}(-\sqrt{2})^{m}$$

and since we must have $m-2 \geq 0$ in the inductive step, $m > 1 = n_0$ so we must establish the base case for $n = 0$ and $n = 1$ (which are both true). In general, one can prove in a similar way that the number of moves is

$$2^{\lceil n/k \rceil} - 1 = 2^{n/k} \sum_{s=0}^{k-1} c_s e^{i2\pi ns/k} - 1$$

for some appropriate values of $c_0, \ldots, c_{k-1}$.

Observe that $2^{\lceil n/k \rceil}$ is infinitely faster than 2^n, in fact the ratio $2^{\lceil n/k \rceil}/2^n$ is asymptotically equal to $2^{-n(k-1)/k}$, which approaches 0 for large n (and a fixed k). By choosing $k \approx n/\log_2 f(n)$, where $1 < f(n) \leq 2^n$, the number of moves for this version of the Tower of Hanoi grows asymptotically as $f(n)$.

Finally, one interesting aspect of this variation is that the number of optimal solutions can be huge. If we denote this number by b_n, for n disks, and let $r = n \bmod k$, then

$$b_n = 1 + \sum_{i=r+1+k\cdot 0^r}^{\min\{k,n\}} b_{n-i}^2$$

Therefore, $b_n = 1$ iff $n \bmod k = 0$. If $n \bmod k = k - 1 \neq 0$, then $b_n = 1 + b_{n-k}^2$ (and $b_{k-1} = 1$), so this generates the sequence $1, 2, 5, 26, 677, 458330, \ldots$ for $n \equiv k-1$, e.g. for odd n when $k = 2$, which can be shown to grow asymptotically as $(1.225902\ldots)^{2^{(n+1)/k}}$ (https://oeis.org/A003095 [10]).

IV. Rubber Disk in The Way

In this variation, and in addition to the stack of n disks, there is a rubber disk initially placed through one of the two other pegs as shown in Figure 4. The rubber disk is *rubbery* and light so it can sit on any disk, but only disks $1, \ldots, k$ where $k \in \{0\} \cup \mathbb{N}$ can appear above the rubber disk (when $k = 0$ no disk can sit on top of the rubber disk). At any point in time, however, all disks must represent a legitimate Tower of Hanoi state, i.e. respecting proper placement of disk sizes once the rubber disk has been ignored and taken out of the picture. The goal of this variation, called Rubber Disk in The Way, is to transfer the entire stack of disks to another peg and end up with the rubber disk on its original peg (with nothing on top or below).

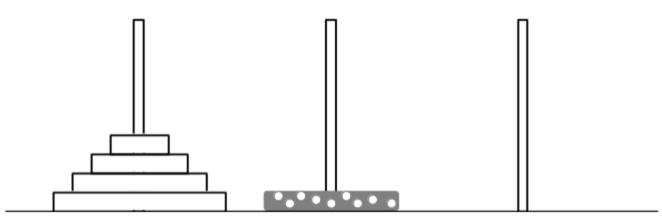

Fig. 4. Rubber disk in the way, $n = 4$

It is not immediately obvious how one could benefit from placing a disk on top of the rubber disk (e.g. when $k > 0$). For instance, a trivial solution, though not optimal since it ignores k, is to first move the rubber disk on top of the initial stack of height n, then treat the resulting problem as one instance of Tower of Hanoi with $n+1$ disks, where the rubber disk plays to role of disk 1 (the smallest). Finally, the rubber disk (still on top of the stack) is moved to its original peg. This explicitly requires $1 + (2^{n+1} - 1) + 1 = 2^{n+1} + 1$ moves (which can still be optimized because the first and last moves of the rubber disk may be redundant, so to be exact, we have $2(2^n - 1) - 1$ and $2(2^n - 1) + 1$ moves for odd and even n, respectively). Since the above solution makes no use of k (in fact it treats k as 0), then can we do better? Well, if $k \geq n - 1$, then we can simply transfer the stack of n disks in $2^n - 1$ moves with the standard Hanoi algorithm while keeping the rubber disk in place at all times. Therefore, we must use k somehow, and the optimal number of moves will vary asymptotically in $[2^n, 2 \cdot 2^n]$.

We first present a non-optimal algorithm that guarantees an asymptotic $(2 - \frac{1}{2^{\alpha-1}})2^n$ number of moves, where $0 < \alpha = n - k \leq n$. To keep the illustration simple, we assume that the original stack of n disks will end up on some peg and the rubber disk on another (not necessarily its original peg). As this can be fixed by at most two additional moves, [2] the asymptotic behavior is preserved. In addition, we use n as an argument within the recursive function RubberDiskInTheWay, as well as a global parameter (in Forward).

```
RubberDiskInTheWay(n, x, y, z)
  if n > k
     then Hanoi(k, x, z, y)
          (y, z) ←Forward(k + 1, x, y, z)
          Move(1, x, z)
          (x, y) ←Backward(n − 1, x, y, z)
          Hanoi(k, y, x, z)
     else Hanoi(n, x, y, z)
```

```
Forward(h, x, y, z)
  if h < n
     then Move(1, x, z)
          Hanoi(h, y, x, z)
          return Forward(h + 1, x, z, y)
  return (y, z)
```

```
Backward(h, x, y, z)
  if h > k
     then Hanoi(h, y, z, x)
          Move(1, y, z)
          return Backward(h − 1, y, x, z)
  return (x, y)
```

The algorithm above works by first placing the top k disks on the rubber disk (first call to Hanoi in RubberDiskInTheWay) to make a stack of height $k+1$, then gradually grow the height of that stack to n (using Forward) until disk n is free to move. After moving disk n, we gradually shrink the height of the stack from n down to $k+1$ (using Backward) to pile up the $n-k$ largest disks (thus moving $n-k-1$ disks on top of disk n). Finally, we transfer the k disks that sit above the

[2] An initial move of the rubber disk to the other empty peg will produce the symmetric solution. In addition, one last move of the rubber disk can ensure its proper positioning.

rubber disk (second call of Hanoi in RubberDiskInTheWay), leaving the rubber disk free.

It is easy to see that RubberDiskInTheWay contributes asymptotically $2 \cdot 2^k$ moves through its two calls to Hanoi, and

$$[1+(2^{k+1}-1)]+[1+(2^{k+2}-1)]+\ldots+[1+(2^{n-1}-1)]$$

moves through each of the Froward and Backward algorithms, resulting in a total of

$$2(2^k+2^{k+1}+\ldots+2^{n-1}) = 2^{k+1}+2^{k+2}+\ldots+2^n$$

moves (asymptotically). Perhaps one of the famous proofs by induction pertains to power series, so we can easily prove (by induction) our result stated earlier for $n > k$ (recall $\alpha = n-k$).

$$2^{k+1}+\ldots+2^n = \Big(2-\frac{1}{2^{\alpha-1}}\Big)2^n$$

The inductive step for $n = m > n_0$ proceeds as follows:

$$2^{k+1}+\ldots+2^m = (2^{k+1}+\ldots+2^{m-1})+2^m$$

$$= \Big(2-\frac{1}{2^{m-1-k-1}}\Big)2^{m-1}+2^m = 2^m+2^m-\frac{2^{m-1}}{2^{m-1-k-1}} =$$

$$2\cdot 2^m - \frac{2^m}{2^{m-k-1}} = \Big(2-\frac{1}{2^{m-k-1}}\Big)2^m$$

Now let us articulate the base case. Since we had to isolate one term in the above sum (namely 2^m), the inductive step should work as long as the sum has at least that one term and, therefore, m must satisfy $m \geq k+1$. So $m > k = n_0$ and hence we must verify the base case for $n = k$. But since the statement of the proven property requires $n > k$, such a base case is not valid. How do we handle this subtlety? Well, the inductive step still works if $m > k+1$, so we could set $n_0 = k+1$ and verify the base case for all $n \leq k+1$. But since $n > k$, the base case for $n = k+1$ is all we need, which states that $2^{k+1} = (2-\frac{1}{2^{1-1}})2^{k+1}$ (true).

In light of the above discussion, when a condition like $n > k$ is not stated explicitly, do we face an endless search for a base case? The answer is, of course, "No" because one has to know something about what is being proved. A statement such as the above requires a condition for it to be true. On the other hand, if the truth of it cannot be established, then the failure of the base case is exactly what we want.

One can interpret the solution presented above as wedging the rubber disk en route in its "correct" relative position below the top k and above the remaining $n-k$ disks. Therefore, we seem to be solving for a setting in which disk $k+1$ was magically pulled away from the stack and placed on a separate peg, and must remain there after the transfer. Intuitively, pulling the smallest disk away does not affect the asymptotic number of moves, while pulling the largest disk away reduces that asymptotic number by half (with the general number of moves being anywhere between the two bounds).

However, the algorithm still does not benefit from the fact that the rubber disk itself can be placed anywhere. Therefore, we can do even better! In fact, the optimal solution is not that hard to conceive. The trick is to virtually consider the rubber disk and the smallest k disks as one entity (which can assume two configurations, either rubber disk on top of k disks, or k disks on top of rubber disk). This entity represents the smallest disk in a Tower of Hanoi instance with $n-k+1$ disks, where this smallest disk requires $1+(2^k-1) = 2^k$ moves to move once. By Fact 3, it is then easy to see that the total number of moves is asymptotically $2^k(2^{n-k})+2^{n-k} = 2^n+2^{n-k}$, since the smallest disk must move 2^{n-k} times. Therefore, the optimal number of moves is asymptotically $(2-\frac{2^k-1}{2^k})2^n$.

We end this section with a funny recursive algorithm for the Rubber Disk in The Way variation, for the sake of illustrating how a "seemingly good" solution might not work out nicely after all:

```
KeepMovingIt(n, x, y, z)
  if n > 0
     then Move(1, y, z) (rubber)
          KeepMovingIt(n − 1, x, z, y)
          Move(1, z, y) (rubber)
          Move(1, x, z)
          Move(1, y, x) (rubber)
          KeepMovingIt(n − 1, y, x, z)
          Move(1, x, y) (rubber)
```

There is a nice symmetry to the solution and, in addition, observe that by ignoring the rubber moves in the above description, the algorithm will be exactly that of a standard Tower of Hanoi. Unfortunately, the recurrence $a_n = 2a_{n-1}+5$ is not as good. By changing the recurrence into a homogeneous one (with the same technique used so far), we obtain $a_n = 3a_{n-1}-2a_{n-2}$, with the characteristic equation $x^2 = 3x-2$, and $r_1 = 1$ and $r_2 = 2$ as the two distinct roots. Therefore, we conclude that $a_n = c_1 + c_2 2^n$. Now,

$$a_0 = c_1 + c_2 = 0$$

$$a_1 = c_1 + 2c_2 = 1$$

$$a_2 = c_1 + 4c_2 = 7$$

where a_0 and a_1 correspond to trivial solutions, and a_2 is obtained from the recurrence $a_2 = 2a_1 + 5 = 7$ (already an indication that our solution is not optimal). A common mistake here is to use a_0 and a_1 to solve for c_1 and c_2, and obtain $a_n = 2^n - 1$ (as good as plain old Hanoi!). Indeed, a_0 and a_1 cannot be used as the base to solve for c_1 and c_2 because $a_1 \neq 2a_0 + 5$, a result of the solution not being optimal (the same is true for a_0 and a_2 since $a_2 \neq 3a_1 - 2a_0$). Therefore, we should use a_1 and a_2 instead, to obtain $c_1 = -5$, $c_2 = 3$, and $a_n = 3\cdot 2^n - 5$ (and observe that this does not satisfy a_0). This is outside the asymptotic range $[2^n, 2\cdot 2^n]$, as expected. The sequence $a_{n\geq 1} \bmod 10$ cycles through 1, 7, 9, and 3 (same as DoubleDecker but shifted), which can be easily proved by induction (Hanoi cycles through 1, 3, 7, and 5).

V. Exploding Tower of Hanoi

We now consider an Exploding Tower of Hanoi. In this variation, if the largest remaining disk becomes free with nothing on top, it explodes and disappears. The goal is to make the whole tower disappear. For instance, $a_n = 0$ when $n \leq 1$ (with either no disks or one free disk). With two disks, once the smallest is moved, the largest disk becomes free and explodes, so the smallest, being now the largest remaining free disk, will follow, resulting in $a_2 = 1$. Similarly, it is not hard to see that $a_3 = 2$. The optimal solution can be derived as follows: To free the largest disk, one must move the second largest, which as illustrated for the case of $n = 2$, will also explode. Observe that no disks can explode prior to the largest. Therefore, we first transfer $n-2$ disks to some peg, then move the second largest disk to another, hence freeing two disks for two explosions at once, and finally repeat the solution for the remaining $n-2$ disks.

```
Exploding(n, x, y, z)
   if n > 1
       then Hanoi(n − 2, x, z, y)
               Move(1, x, z)
               disks n and n − 1 explode
               Exploding(n − 2, y, x, z)
```

Given this algorithm, we establish the recurrence:

$$a_n = a_{n-2} + 2^{n-2}$$

and change it into a homogeneous one by annihilation of 2^{n-2} as follows:

$$a_n = a_{n-2} + 2^{n-2}$$

$$2 \cdot a_{n-1} = 2 \cdot a_{n-3} + 2 \cdot 2^{n-3}$$

$$a_n - 2a_{n-1} = a_{n-2} - 2a_{n-3}$$

to finally obtain $a_n = 2a_{n-1} + a_{n-2} - 2a_{n-3}$ and the characteristic equation $x^3 = 2x^2 + x - 2$. By observing that $r_1 = 1$ is a root, we express the characteristic equation as $(x-1)(x^2 - x - 2) = 0$ and solve the quadratic equation for the other two roots. The three distinct roots will be $r_1 = 1$, $r_2 = -1$, and $r_3 = 2$. Therefore, $a_n = c_1 + c_2(-1)^n + c_3 2^n$, and since

$$a_0 = c_1 + c_2 + c_3 = 0$$

$$a_1 = c_1 - c_2 - 2c_3 = 0$$

$$a_2 = c_1 + c_2 + 4c_3 = 1$$

we have $c_1 = -1/2$, $c_2 = 1/6$, and $c_3 = 1/3$. Finally,

$$a_n = \frac{(-1)^n + 2^{n+1} - 3}{6}$$

So the Exploding Tower of Hanoi is asymptotically three times as fast as the Tower of Hanoi. An interesting aspect of this solution, and more generally solutions to recurrences for integer sequences, is the ability to generate statements related to divisibility that are suitable for proofs by induction. For instance, an immediate thought is to prove (by induction) that $(-1)^n + 2^{n+1} - 3$ is a multiple of 6, as follows for $n = m > n_0$:

$$(-1)^m + 2^{m+1} - 3 = (-1)^{m-2} + 4 \cdot 2^{m-1} - 3$$

$$= [(-1)^{m-2} + 2^{m-1} - 3] + 3 \cdot 2^{m-1} = 6k + 6 \cdot 2^{m-2}$$

The above (strong) induction requires that 2^{m-2} is an integer so $m - 2 \geq 0$, and thus $m > 1 = n_0$. So we must verify the base case for $n = 0$ and $n = 1$, both of which are true.

Iterating $a_{n\geq 0}$ produces the following integer sequence $0, 0, 1, 2, 5, 10, 21, 42, 85, 170, \ldots$ (https://oeis.org/A000975 [10]), with $a_n = 2a_{n-1}$ if n is odd, and $a_n = 2a_{n-1} + 1$ if n is even (and $n > 0$). This property can be proved by induction using the recurrence we derived earlier. The inductive step for $n = m > n_0$ works as follows:

$$a_m = 2a_{m-1} + a_{m-2} - 2a_{m-3} = 2a_{m-1} + [a_{m-2} - 2a_{m-3}]$$

which is $2a_{m-1} + 0$ if $m - 2$ (and m) is odd, and $2a_{m-1} + 1$ if $m - 2$ (and m) is even. Since the inductive step requires that $m - 3 \geq 0$, i.e. $m > 2 = n_0$, we must verify the base case for $n = 1$ and $n = 2$, which are both true since $a_1 = 2a_0$ and $a_2 = 2a_1 + 1$.

The above property means that $a_{n\geq 1}$ is the n^{th} non-negative integer for which the binary representation consists of alternating bits. Therefore, when $n \geq 2$, this alternation starts with 1 and has exactly $n - 1$ bits, making it super easy to write down a_n in binary (for Hanoi, a_n consists of n bits that are all 1s). [3] This is another nice candidate for a proof by induction, for $n = m > n_0$:

The number a_{m-1} has $m - 2$ alternating bits starting with 1. If m is odd (so is $m - 2$), then a_{m-1} has an odd number of bits thus ending with 1, and $a_m = 2a_{m-1}$ shifts the bits of a_{m-1} and adds 0. If m is even (so is $m-2$), then a_{m-1} has an even number of bits thus ending with 0, and $a_m = 2a_{m-1} + 1$ shifts the bits of a_m and adds 1. In both cases, a_m has $m - 1$ alternating bits starting with 1.

The working of this inductive step relies on a_{m-1} having at least one 1 bit in its binary representation; therefore, we need $m - 2 \geq 1$ or $m > 2 = n_0$. This means $n = 2$ must be our base case and, indeed, $a_2 = 1$ has an alternating pattern of $2 - 1 = 1$ bit starting with 1.

Finally, it is worth noting here that this is one of the few Tower of Hanoi variations where the optimal number of moves can be even ($a_{n\geq 1} \bmod 10$ cycles through 0, 1, 2, and 5).

VI. The Pivot

In this variation, called the Pivot Tower of Hanoi, only two types of moves will be allowed. Either the smallest disk (disk 1) is moved to any peg, or some disk and the smallest exchange places (and this is considered to be one move). Therefore, except for the smallest disk, disks can only move by *pivoting* around disk 1, hence the name of this variation. Of course, we still require that, by pivoting, a disk cannot be placed on a smaller one. So, for instance, only the disk on top of a stack can be exchanged with the smallest.

It is not immediately clear why this variation can be solved. But it can, since every move of disk $i \neq 1$ to peg x can be emulated by moving disk 1 to peg x then making the exchange with disk i, and in fact, the solution can be even faster than the original Tower of Hanoi; the main focus will be on a proof by induction.

First we observe that the optimal solution for the Tower of Hanoi is an *alternating Hanoi* sequence of moves in which

[3] The recurrence $a_n = a_{n-2} + 2^{n-2}$ already suggests that a_n is either a sum of consecutive even powers of 2, or a sum of consecutive odd powers of 2, hence the alternating bit pattern.

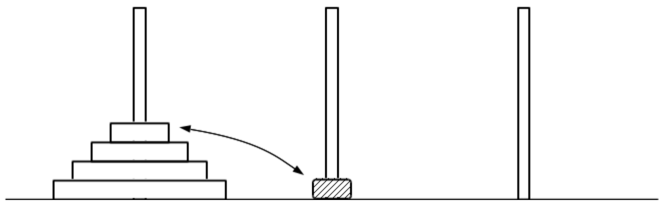

Fig. 5. Pivot Tower of Hanoi for $n = 5$, the disk on top of a stack can be exchanged with the smallest.

the smallest disk is involved in every other move. This can be seen from Fact 3: Since disk 1 makes 2^{n-1} moves, there are $2^{n-1} - 1$ moves of the other disks (for a total of $2^n - 1$ moves). With the solution being optimal, we never move the smallest disk twice in a row, so the 2^{n-1} moves of disk 1 must perfectly interleave the rest, and we have an alternating sequence of moves (starting with the smallest disk). Similarly, an *alternating Pivot* sequence for the Pivot Tower of Hanoi is a sequence of moves that alternate between moving disk 1 (also the starting move) and pivoting a disk (exchanging it with disk 1). We will say that two alternating sequences are *equivalent* if they result in the same placement of all disks, except possibly for the smallest (disk 1). We are now ready for a proof by induction for the following:

(H2P) Every alternating Hanoi sequence of length l has an equivalent alternating Pivot sequence of length at most l, and (P2H) every alternating Pivot sequence of length l has an equivalent alternating Hanoi sequence of length at most l.

An inductive step can proceed as follows for $l = k > l_0$:

(H2P) Given an alternating Hanoi sequence $h_1, \ldots, h_k$, if h_k is a move of disk 1, then consider the alternating Pivot sequence equivalent to $h_1, \ldots, h_{k-1}$; this alternating Pivot sequence of length at most $k - 1$ is also equivalent to $h_1, \ldots, h_k$ (since the last move is for disk 1). If h_k is a move of disk $i \neq 1$ to peg x, then consider the alternating Pivot sequence equivalent to $h_1, \ldots, h_{k-2}$; we can assume that this alternating Pivot sequence of length at most $k - 2$ ends with pivoting, otherwise we can simply drop the last move of disk 1 only to make the sequence even shorter. Therefore, we can extend $p_1, \ldots, p_{k-2}$ by first moving disk 1 to peg x, then exchanging disk i with disk 1 (pivoting), to produce an alternating Pivot sequence equivalent to $h_1, \ldots, h_k$ of length at most $(k - 2) + 2 = k$.

(P2H) Given an alternating Pivot sequence $p_1, \ldots, p_k$, if p_k is a move of disk 1, then consider the alternating Hanoi sequence equivalent to $p_1, \ldots, p_{k-1}$; this alternating Hanoi sequence of length at most $k-1$ is also equivalent to $p_1, \ldots, p_k$ (since the last move is for disk 1). If p_k is an exchange of disk i on peg x with disk 1 on peg y, then consider the alternating Hanoi sequence equivalent to $p_1, \ldots, p_{k-2}$; we can assume that this alternating Hanoi sequence of length at most $k - 2$ ends with a move for some disk $j \neq 1$, otherwise we can simply drop the last move of disk 1 only to make the sequence even shorter. Therefore, we can extend $h_1, \ldots, h_{k-2}$ by first moving disk 1 to peg z (if it's not already there), then moving disk i from peg x to peg y, to produce an alternating Hanoi sequence equivalent to $p_1, \ldots, p_k$ of length at most $(k - 2) + 2 = k$.

Since the (strong) inductive step requires $k - 2 \geq 0$ (the length of the empty sequence), we must have $k > 1 = l_0$ and hence verify the base case for $l = 0$ and $l = 1$, which are both true because the empty sequence (of length zero) is equivalent to any alternating (Hanoi or Pivot) sequence of length $l \leq 1$, in which only the smallest disk can move.

The equivalence of alternating sequences (we only need H2P) implies that the first $2^n - 2$ moves (excluding the last move of disk 1) in the optimal solution of the Tower of Hanoi have an equivalent alternating sequence of moves for the Pivot Tower of Hanoi. Adding one last move of the smallest disk positions it correctly on top of the stack for a total of at most $2^n - 1$ moves. But how fast is the optimal solution for the Pivot Tower of Hanoi if we do not require the sequence of moves to be alternating (we have no choice but to alternate in the Tower of Hanoi)?

A quick exploration reveals that, for the Pivot Tower of Hanoi, $a_0 = 0$, $a_1 = 1$, $a_2 = 3$, $a_3 = 7$ (so far matching the Tower of Hanoi), and $a_4 = 13$ (as opposed to $a_4 = 15$ for the Tower of Hanoi). To gain some insight into the recurrence, an optimal solution must transfer the top $n-1$ disks but without placing the smallest on top of the stack, so that it can be used for pivoting to move the largest disk to its destination. This maneuver results in a stack of $n - 2$ disks with disk 1 separated from the stack, which implies that the solution for $n - 1$ disks needs to be repeated, except that the first move is now provided for free! This suggests that the recurrence is $a_n = (a_{n-1} - 1) + 1 + (a_{n-1} - 1) = 2a_{n-1} - 1$. However, depending on the value of n, the separation of the smallest disk might not place it on a favorable peg; for instance, disk 1 can end up sitting on top of disk n (with disks $2, \ldots, n-1$ forming a stack). So we require an extra move before we can pivot disk n to its destination. But this will place disk 1 back on the same peg, so we require yet another extra move to carry out the remainder of the solution. As it turns out for $n \geq 3$, $a_n = 2a_{n-1} - 1$ when n is even, and $a_n = 2a_{n-1} - 1 + 2 = 2a_{n-1} + 1$ when n is odd.

```
Pivot(n, x, y, z, NoLstMv = False, FrstMvFree = False)
  if n > 2
     then Pivot(n − 1, x, z, y, True, FrstMvFree)
          if n ≡ 1 (mod 2)
             then Move(1, x, z)
          Exchange(n)
          if n ≡ 1 (mod 2)
             then Move(1, x, z)
          Pivot(n − 1, y, x, z, NoLstMv, True)
  if n = 2
     then if not FrstMvFree
             then Move(1, x, z)
          Exchange(n)
          if not NoLstMv
             then Move(1, x, z)
  if n = 1
     then if not NoLstMv and not FrstMvFree
             then Move(1, x, z)
```

To annihilate the extra term in the non-homogeneous recurrence, we find

$$a_n + a_{n-1} = 2a_{n-1} + 2a_{n-2}$$

to yield $a_n = a_{n-1} + 2a_{n-2}$ and the characteristic equation $x^2 = x + 2$ with roots $r_1 = 2$ and $r_2 = -1$. So $a_n = c_1 2^n + c_2(-1)^n$. Since our recurrence works for $n \geq 3$, we now have:

$$a_2 = 4c_1 + c_2 = 3$$

$$a_3 = 8c_1 - c_2 = 7$$

with $c_1 = 5/6$ and $c_2 = -1/3$. Therefore, when $n \geq 2$:

$$a_n = \frac{5 \cdot 2^n - 2(-1)^n}{6}$$

So the Pivot Tower of Hanoi is asymptotically $6/5$ times as fast as the Tower of Hanoi. It is worth noting here that the number of exchanges satisfy, for $n \geq 1$, $e_n = 2e_{n-1} + 1$, with $e_1 = 0$; thus $e_n = 2^{n-1} - 1$ for $n \geq 1$, which means $\lim_{n\to\infty} e_n/a_n = 3/5$, so exchanges make about a $3/5$ fraction of the total number of moves (as opposed to half in the trivial alternating solution).

As in the previous section, the expression for a_n suggests to prove by induction that $5 \cdot 2^n - 2(-1)^n$ is a multiple of 6 when $n \geq 1$. The inductive step for $m > n_0$ proceeds as following:

$$5 \cdot 2^m - 2(-1)^m = 5 \cdot 4 \cdot 2^{m-2} - 2(-1)^{m-2}$$

$$= [5 \cdot 2^{m-2} - 2(-1)^{m-2}] + 15 \cdot 2^{m-2} = 6k + 6 \cdot 5 \cdot 2^{m-3}$$

For this strong induction we need 2^{m-3} to be an integer, so $m > 2 = n_0$. Therefore, $n = 1$ and $n = 2$ must be verified as (and hence are) the base cases.

Finally, we observe that iterating $a_{n\geq 0}$ produces the following integer sequence: $0, 1, 3, 7, 13, 27, 53, 107, 213, 427, 853, 1707, \ldots$ (https://oeis.org/A048573 [10]). Therefore, starting with $n = 2$, $a_n \bmod 10$ alternates between 3 and 7, and starting with $n = 3$, $a_n \bmod 100$ cycles through 7, 13, 27, and 53. These properties can also be proved by induction.

VII. Beam Me Up Scotty! (The Fibonacci Tower)

For our last variation, we consider one called Beam Me Up Scotty, in which disk i for $1 < i < n$ is *teleported* for free between the two disks $i - 1$ and $i + 1$, whenever the former is sitting directly on top of the latter. [4] This seamless teleport, which does not count among the moves, stands behind the borrowed name of this variation from a phrase in popular culture on Star Trek (even though captain James Kirk never really uttered that phrase) [6].

The solution to this variation will relate the number of moves in a Tower of Hanoi game to the Fibonacci numbers. As usual, we must (recursively) transfer the top $n - 1$ disks first, then move the largest disk to its destination, and finally transfer $n - 2$ disks (recursively) on top of the largest; the $(n-1)^{\text{st}}$ disk will benefit from the free teleport between disks $n - 2$ and n. Obviously, when $n \leq 2$, no disk can benefit from any teleport. The algorithm is shown below, and gives the recurrence $a_n = a_{n-1} + a_{n-2} + 1$, for $n > 2$.

```
BeamMeUpScotty(n, x, y, z)
  if n > 2
     then BeamMeUpScotty(n − 1, x, z, y)
          Move(1, x, z)
          BeamMeUpScotty(n − 2, y, x, z)
          disk n − 1 will be teleported
     else Hanoi(n, x, y, z)
```

Iterating $a_{n\geq 0}$ produces the integer sequence $0, 1, 3, 5, 9, 15, 25, 41, \ldots$ that, starting with a_1, coincides with Leonardo numbers (https://oeis.org/A001595 [10]). For $n > 0$, we can show that $a_n = 2F_{n+1} - 1$, by induction for $n = m > n_0$:

$$a_m = a_{m-1} + a_{m-2} + 1 = 2F_m - 1 + 2F_{m-1} - 1 + 1$$

$$= 2(F_m + F_{m-1}) - 1 = 2F_{m+1} - 1$$

Since the recurrence is defined for $n > 2$, the above (strong) inductive step requires $m > 2$ (also $m - 2 \geq 0$ so that a_{m-1}, a_{m-2}, and F_{m-1} are all defined). Therefore, $m > 2 = n_0$, and the base case must be verified for $n = 1$ and $n = 2$. Indeed, $a_1 = 1 = 2F_2 - 1$ and $a_2 = 3 = 2F3 - 1$.

Solving the recurrence $a_n = a_{n-1} + a_{n-2} + 1$ using $a_n - a_{n-1} = a_{n-1} - a_{n-3}$ and the characteristic equation $x^3 = 2x^2 - 1$, which is $(x - 1)(x^2 - x - 1) = 0$, will result in

$$a_n = \frac{2}{\sqrt{5}}\left[\phi^{n+1} - (1 - \phi)^{n+1}\right] - 1$$

for $n > 0$ and can also be proved by induction, where $\phi = (1+\sqrt{5})/2$ is the golden ratio (in fact this solution is exactly the expression for $2F_{n+1} - 1$ obtained by solving the recurrence for Fibonacci numbers).

Interestingly, one can easily show that the number of teleports also satisfies the recurrence $t_n = t_{n-1} + t_{n-2} + 1$ for $n > 2$ with $t_0 = t_1 = t_2 = 0$, giving $t_n = F_n - 1$ for $n > 0$. Therefore, if we were to add the number of moves and the number of teleports for $n > 0$ we obtain $a_n + t_n = 2F_{n+1} - 1 + F_n - 1 = F_{n+3} - 2$.

Another interesting observation is that the solution to this variation is infinitely faster (in the asymptotic sense) than the original Tower of Hanoi (a feature that is shared only with Move One Get Some Free variation that can explicitly move multiple disks simultaneously). This can be seen from the fact that $a_n \approx -1 + 2\phi^{n+1}/\sqrt{5}$ for large n and that

$$\lim_{n\to\infty} \frac{2\phi^{n+1}}{\sqrt{5} \cdot 2^n} = 0$$

It is worth noting here that if disk n is also allowed to be teleported under disk $n - 1$ whenever disk $n - 1$ makes a move, then $a_{n\geq 0}$ becomes shifted as follows: $0, 1, 1, 3, 5, 9, 15, 25, 41, \ldots$. This gives $a_n = 2F_n - 1$ (and $t_n = t_{n-1} + t_{n-2} = F_{n-1}$) for $n > 0$, which essentially

[4] We do not explicitly require that the teleported disk be on top of its stack; however, this is surprisingly the only possible scenario: when disk $i - 1$ is placed on top of disk $i+1$, the teleported disk i is either free to move (on top of its stack), or sitting directly under disk $i-2$; the latter case is impossible since disk $i - 1$ would have been first to teleport prior to its move.

amounts to solving an instance of $n-1$ disks as originally defined for Beam Me Up Scotty, with one additional free teleport for the largest disk (disk n). This results in $F_{n+2}-1$ for $a_n + t_n$ when $n > 0$. This modification preserves the asymptotic speed of Beam Me Up Scotty relative to the other variations (it speedups it up by ϕ). Regardless, the asymptotic ratio of the number of moves to the number of teleports is $2\phi = 1+\sqrt{5}$.

VIII. On The Speed of Tower of Hanoi

To appreciate of effect of a_n on the time needed to solve a particular variant of the Tower of Hanoi, observe that it will only take about 544 millenniums to solve Beam Me Up Scotty with $n = 64$ disks, compared to the 585 billion years for Tower of Hanoi with the same number of disks, a speedup of more than a million! But the Move One Get Some Free remains the fastest, with a speedup of more than four billions when $n = 64$ and $k = 2$, thus taking about 146 years for Move One Get One Free.

Finally, and for the sake of pointing out a variation with an infinite slow down, consider the original Tower of Hanoi except that every move must involve the middle peg. This variation is mentioned in [3]. Let the name of it be Man in the Middle. A solution is presented below:

```
ManInTheMiddle(n, x, y, z)
  if n > 0
     then ManInTheMiddle(n − 1, x, y, z)
          Move(1, x, y)
          ManInTheMiddle(n − 1, z, y, x)
          Move(1, y, z)
          ManInTheMiddle(n − 1, x, y, z)
```

The recurrence generated by the above algorithm is $a_n = 3a_{n-1} + 2$, which by inspection admits the solution $a_n = 3^n - 1$ (proof by induction), with an asymptotic relative time of $(3/2)^n$ (more than 10^{14} billion years for $n = 64$ disks).

Table 1 lists all variations in decreasing order of their asymptotic time relative to Hanoi (from slowest to fastest).

Man in Middle	k-Decker	Rubber Disk	Hanoi
$(3/2)^n$	$2k$	$1+2^{-k}$	1
10^{23}	$234\cdot10^{10}$ $k=2$	$117\cdot10^{10}$ $k=0$	$585\cdot10^{9}$

Pivot	Exploding	Beam Me Up	Get $k-1$ Free
$5/6$	$1/3$	$\frac{2\phi}{\sqrt{5}}(2/\phi)^{-n}$	$2^{-n(k-1)/k}$
$488\cdot10^{9}$	$195\cdot10^{9}$	$544\cdot10^{3}$	146 $k=2$

TABLE I
ASYMPTOTIC TIME RELATIVE TO HANOI, AND APPROXIMATE REAL TIME IN YEARS NEEDED TO TRANSFER $n = 64$ DISKS, ASSUMING ONE SECOND PER MOVE.

IX. Final Remarks

We explore the Tower of Hanoi as a vehicle to convey classical ideas of recurrences and proofs by induction in a new way. The variations considered here are relatively simple compared to the more research inclined type of problems, and provide a framework to strengthen the understanding of recurrences and mathematical induction via a repetitive and systematic treatment of the subject, while pointing out how to avoid the pitfalls. We summarize below some of the goals/highlights of the approach:

- Strengthen the general understanding of recurrences and proofs by induction.
- Provide a mechanism that teaches how to establish recurrences and think about them (and eventually solve them).
- Describe a systematic way to handle recurrences that is reasonable for introductory discrete mathematics.
- Suggest ways to enrich the standard learning environment e.g. by asking for programming variations to a classically known recursive algorithm.
- Construct proofs by induction from the expressions obtained for solutions to recurrences, and/or by solving a recurrence in different ways and equating the results.
- Highlight the pitfalls that are typically encountered in recurrences (boundary conditions) and proofs by induction (base cases), e.g. by making a clear post-treatment of the base cases in light of the inductive step.
- Create opportunities to have fun with the endless variations of the Tower of Hanoi while learning the concepts.

References

[1] T. Bousch. La tour de stockmeyer. *Séminaire Lotharingien de Combinatoire*, 77:B77d, 2017.

[2] T. Bousch et al. La quatrième tour de hanoï. *Bulletin of the Belgian Mathematical Society-Simon Stevin*, 21(5):895–912, 2014.

[3] R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu. Concrete mathematics: a foundation for computer science. *Computers in Physics*, 3(5):17–18, 1989.

[4] C. Grosu. A new lower bound for the towers of hanoi problem. *arXiv preprint arXiv:1508.04272*, 2015.

[5] A. M. Hinz, S. Klavžar, U. Milutinović, C. Petr, and I. Stewart. *The Tower of Hanoi-Myths and Maths*. Springer, 2013.

[6] J. T. Kirk. (fictional character), star trek. *The original series*, 1966.

[7] U. Levy. Magnetic towers of hanoi and their optimal solutions. *arXiv preprint arXiv:1011.3843*, 2010.

[8] É. Lucas. Récréations mathématiques, vol. iii, gauthier-villars, paris, 1893. *Reprinted several times by Albert Blanchard, Paris.*

[9] G. S. Lueker. Some techniques for solving recurrences. *ACM Computing Surveys (CSUR)*, 12(4):419–436, 1980.

[10] N. J. A. Sloane. The on-line encyclopedia of integer sequences. *published electronically at https://oeis.org*, 2018.

[11] P. K. Stockmeyer. Variations on the four-post tower of hanoi puzzle. In *Congress. Numer.*, volume 102, pages 3–12, 1994.

[12] P. K. Stockmeyer, C. Douglass Bateman, J. W. Clark, C. R. Eyster, M. T. Harrison, N. A. Loehr, P. J. Rodriguez, and J. R. Simmons III. Exchanging disks in the tower of hanoi. *International journal of computer mathematics*, 59(1-2):37–47, 1995.

[13] P. K. Stockmeyer and F. Lunnon. New variations on the tower of hanoi. In *Proc. of International Conference on Fibonaci Numbers and Their Applications*. Citeseer, 2008.

[14] Wolfram|Alpha. *Wolfram Alpha LLC, https://www.wolframalpha.com/input/?i=r\%5E5\%3D2*, 2018.