

# The number of moves of the largest disc in shortest paths on Hanoi graphs

Simon Aumann

Department of Mathematics  
University of Munich  
Munich, Germany  
aumann@math.lmu.de

Katharina A. M. Götz

German Space Operations Center  
German Aerospace Center  
Weßling, Germany  
katharina.goetz@dlr.de

Andreas M. Hinz

Department for Mathematics  
and Computer Science  
University of Maribor  
Maribor, Slovenia  
Department of Mathematics  
University of Munich  
Munich, Germany  
hinz@math.lmu.de

Ciril Petr

Department for Mathematics  
and Computer Science  
University of Maribor  
Maribor, Slovenia  
petr@iskratel.si

Submitted: 2014-04-04; Accepted: 2014-11-05; Published: XX  
Mathematics Subject Classifications: 05C12, 68W05, 05-04

## Abstract

In contrast to the widespread interest in the *Frame-Stewart Conjecture* (FSC) about the optimal number of moves in the classical Tower of Hanoi task with more than three pegs, this is the first study of the question of investigating shortest paths in *Hanoi graphs*  $H_p^n$  in a more general setting. Here  $p$  stands for the number of pegs and  $n$  for the number of discs in the Tower of Hanoi interpretation of these graphs. The analysis depends crucially on the number of *largest disc moves* (LDMs). The patterns of these LDMs will be coded as binary strings of length  $p - 1$  assigned to each pair of starting and goal states individually. This will be approached both analytically and numerically. The main theoretical achievement is the existence, at least for all  $n \geq p(p-2)$ , of optimal paths where  $p-1$  LDMs are necessary. Numerical results, obtained by an algorithm based on a modified breadth-first search making use of symmetries of the graphs, lead to a couple of conjectures about some cases not covered by our ascertained results. These, in turn, may shed some light on the notoriously open FSC.

**Keywords:** Tower of Hanoi, Hanoi graphs, shortest paths, symmetries, breadth-first search

# 1 Introduction

The general task of the Tower of Hanoi (TH) puzzle reads as follows. There are  $p \geq 3$  pegs,  $n \geq 1$  discs and two regular distributions of the discs, i.e. no disc lies on a smaller one, on the pegs. A first task is to find any path from the first to the second state just using legal moves, i.e. transfers of single topmost discs obeying the *divine rule* never to destroy regularity.

The question of *shortest* paths between any two states of the puzzle is still the outstanding riddle about the puzzle 130 years after its invention by the French number theorist Édouard Lucas (1842–1891) [3],<sup>1</sup> who asked for such an optimal solution between two *perfect* states, i.e. states with all discs on one peg. A crucial role is played by the largest disc, because its moves are most obstructed by the positions of the other discs and the divine rule. In fact, it can only move if it is alone on its peg, another peg (its goal in the move) is empty and consequently all other discs lie on pegs different from these two.

The proof for the fact that in a perfect-to-perfect task for the Tower of Hanoi with 3 pegs and  $n \in \mathbb{N}$  discs the largest disc moves precisely once carries over to  $p > 3$  pegs (cf. [2, p. 119]): before the first move of disc  $n$ , a tower of  $n - 1$  discs has to be distributed on  $p - 2$  pegs, after the last move of disc  $n$ , some distribution of  $n - 1$  discs on  $p - 2$  pegs has to be collected on one different peg; so first and last move must coincide in an optimal path and any second move of disc  $n$  would produce a path which is strictly longer than the minimal one. Contrary to the case  $p = 3$  (see [12, Theorem 2.7]), however, it is a priori not even clear whether the largest disc moves only once for larger  $p$  if just the goal peg is perfect.

Nevertheless, it has been assumed for a long time that the largest disc moves only *once* in a shortest path from one state to another [24]. So it came to some surprise when it was pointed out in [19] and [9] that there are shortest paths where the largest disc moves or even has to move twice. (This fact had already been observed in the literature of psychology though; see [15, 16].) Meanwhile, it has been found that in fact for every state of the classical three-pegs-TH, with the exception of the perfect states, there are shortest paths to some other states including two *largest-disc-moves* (LDMs); see [11, Corollary 3.6]. The decision problem, whether or not for given initial and goal states two LDMs are appropriate has been successfully dealt with using a finite automaton [21].

No provably correct efficient algorithm is known for the problem to find a shortest path between two states of the TH with more than three pegs, a task which has been posed for the first time by H. E. Dudeney in 1902 [6, p. 367f] in the perfect-to-perfect setting. The strategies for this problem devised by J. S. Frame [7] and B. M. Stewart [23], which have later been found to be equivalent with respect to move numbers (cf. [18]), have not been proven to be optimal until today. As long as provably optimal strategies to solve the more general *regular to regular* problem are missing, the only way to study LDMs is by an exhaustive search of shortest paths in the corresponding *Hanoi graphs*

---

<sup>1</sup>For more historical background, see [12, Chapter 0]

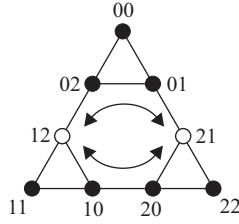


Figure 1: A pair of vertices linked by two shortest paths

$H_p^n$ , i. e. the state graphs representing the TH with  $p \geq 3$  pegs and  $n \in \mathbb{N}$  discs. In Figure 1 we see an example on  $H_3^2$  with a pair of vertices joined by two optimal paths, one with one LDM and the other one with two. The vertices of  $H_p^n$  are written in the form  $r = r_n r_{n-1} \dots r_1 \in [p]_0^n$ , where the value of  $r_d \in [p]_0$  denotes the label of the peg disc  $d \in [n]$  is lying on in state  $r \in V(H_p^n) = [p]_0^n$ ; we write  $[q]$  for  $\{1, \dots, q\}$  and  $[q]_0$  for  $\{0, \dots, q-1\}$ . The edge set of  $H_p^n$  is

$$E(H_p^n) = \{ \{ \underline{s}i\bar{s}, \underline{s}j\bar{s} \} \mid i, j \in [p]_0, i \neq j, d \in [n], \underline{s} \in [p]_0^{n-d}, \bar{s} \in ([p]_0 \setminus \{i, j\})^{d-1} \} .$$

Here an edge represents the move of disc  $d$  between pegs  $i$  and  $j$ . We will call  $p$  the *base* and  $n$  the *exponent* of the Hanoi graph  $H_p^n$ , since the number of vertices is obviously  $|H_p^n| = p^n$ ; cf. [12, Section 2.3].

The upper bound for the number of LDMs in a shortest path is based on the rule “they never come back”:

**Theorem 1.1.** *The number of LDMs in a shortest path in  $H_p^n$  is at most  $p - 1$ .*

*Proof.* In a shortest path the largest disc will never return to a peg it has left before: all the LDMs in between could be deleted, because the moves of the other discs are not obstructed by the position of disc  $n$ . Therefore,  $k$  LDMs will involve  $k + 1$  different pegs, whence  $k \leq p - 1$ .  $\square$

Already for  $H_3^3$  we have an example that two LDMs are *necessary* in a *minimal solution*, i.e. to get from some state to another one in the minimum number of moves. This can be seen in Figure 2.

In order to be able to specify which set of pairs of start and target states has to be analyzed, we define a *task* as an ordered pair  $(s, t)$  of distinct states  $s$  (the initial state) and  $t$  (the goal state); we also write  $s \rightarrow t$ . There are  $p^n (p^n - 1)$  tasks in  $H_p^n$ . Of these, there are  $p^n (p^{n-1} - 1)$  tasks with  $s_n = t_n$ , i.e. with no LDM in a shortest path, because all moves of disc  $n$  could again be left out. Therefore,  $(p - 1)p^{2n-1}$  tasks do have at least one LDM. For these we may assume  $s_n = 0, t_n = p - 1$ , which is one of  $p(p - 1)$  possible cases. So we are investigating  $p^{2(n-1)}$  tasks. This can be further reduced by taking into account the symmetries caused by permutations of the pegs or by inverting the pair of states of a task. (In fact, the automorphism group of  $H_p^n$  is equivalent to the symmetry

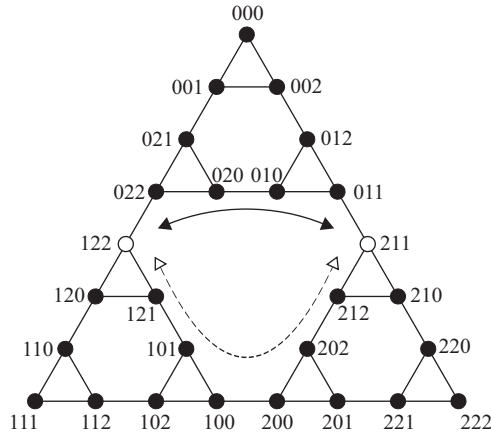


Figure 2: A pair of vertices necessarily needing two LDMs to be linked by an optimal path

group of  $[p]_0$ , i.e. to  $S_p$ ; see [20, Main Theorem] or [12, Theorem 5.33].) The numerical amount of this reduction can be calculated, using Burnside's Lemma [12, Corollary 0.9]. It is even possible, by the Orbit-Stabilizer Theorem [12, Theorem 0.8], to determine the sizes of the corresponding equivalence classes.

Two states are considered to be equivalent if one state emanates from the other by a permutation of the pegs:

$$s \sim s' : \Leftrightarrow \exists \sigma \in S_p : \sigma \circ s = s',$$

where  $\sigma \circ s := \sigma(s_n) \dots \sigma(s_1)$ .

A formula for the number of equivalence classes (also called *equi-sets*) of states on  $H_p^n$  depending on  $p$  and  $n$  can be derived using Burnside's Lemma. For each permutation  $\sigma \in S_p$  we have to calculate the number of states that are left unchanged under this permutation. Obviously, this is true for a state if and only if all discs are placed on the  $q$  non-permuting pegs. Hence, considering a *derangement* (i.e. a permutation without fixed points) of  $p - q$  pegs, there are exactly  $q^n$  fixed states. At first selecting  $q$  from  $p$  pegs, the following formula for the number of equi-sets is obtained:

$$|V(H_p^n)/\sim| = \frac{1}{p!} \sum_{q=1}^p \binom{p}{q} q^n (p-q)! = \sum_{q=0}^p q^n \frac{(p-q)!}{q!(p-q)!}$$

where

$$k! := k! \sum_{j=0}^k \frac{(-1)^j}{j!}$$

is the *subfactorial* of  $k$ , representing the number of derangements on  $[k]$  (cf. [12, Exercise 0.3]). In a similar fashion the number of equi-sets of tasks can be determined; cf. [8].

Let us describe an adroit algorithm to pick one representative state (with the additional property  $s_n = 0$ ) out of each equivalence class. We start to construct a tree with root 0 for the position of the largest disc. At every successive level  $k$  we concatenate to the right a position  $s_{n-k}$  of disc  $n - k$  to each word in level  $k - 1$ , taking care that the new words lie in different equi-sets. Since the  $q$  non-empty pegs are distinguishable by the assignment of the discs  $n, \dots, n - k + 1$ , all states where disc  $n - k$  is added to one of these pegs necessarily belong to different equi-sets. In contrast, all states with disc  $n - k$  placed on an empty peg are equivalent by definition. Hence, disc  $n - k$  will be added to all non-empty pegs and to the empty peg with the smallest label. Therefore the number of successors of a vertex is given by  $q + 1$  if  $q < p$  and otherwise by  $q = p$ . When level  $n - 1$  is achieved, the algorithm stops. The tree spanned so far may be stored for further calculations, since adding one disc corresponds to simply adding the next level. The example of  $H_3^4$  is executed in Figure 3.

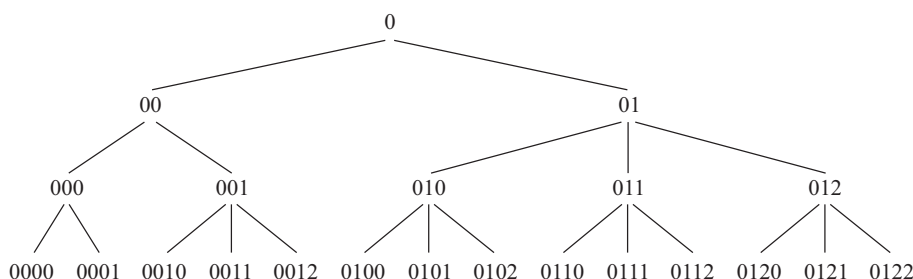


Figure 3: Generating representatives of equivalence classes

In case we want to do calculations for a couple of towers with different numbers of pegs, it should be more effective, at first not to constrict  $p$ , but to construct the whole tree using  $P$  peg labels, where  $P$  stands for the largest peg number one is interested in, and afterwards to sort out those configurations where more than  $p$  pegs are involved.

To classify the different kinds of behavior of tasks with respect to the number of LDMs in shortest paths, let us denote each case by a binary string of  $p - 1$  bits  $b_{p-1} \dots b_1$  called the *bit vector*: we set  $b_i = 1$  iff an optimal solution using  $i$  LDMs exists. So  $0 \dots 0$  signifies the case of no LDM and, e.g.,  $01101$  for  $p = 6$  means that there are shortest paths with 1, 3 or 4 LDMs, but none with 2 or 5. We will also identify these *LDM codes* (or *LDM types*) with their numerical values  $\sum_{i=1}^{p-1} b_i \cdot 2^{i-1}$  in decimal, e.g., 13 in the example.

For the classical case  $p = 3$  we already have a complete quantitative overview. Let us denote by  $x_n$  the solution of the recurrence  $x_0 = 0$ ,  $x_1 = 1$ ,  $x_{n+2} = 5x_{n+1} - 2x_n$  for  $n \in \mathbb{N}_0$ , i.e.  $x_n = \frac{1}{\sqrt{17}} (\Theta_+^n - \Theta_-^n)$ , where  $\Theta_{\pm} := \frac{1}{2} (5 \pm \sqrt{17})$ .

**Proposition 1.1.** *For  $p = 3$  and  $n \in \mathbb{N}$ , the  $9^{n-1}$  tasks  $(s, t)$  from  $H_3^n$  with  $s_n = 0$ ,  $t_n = 2$  decompose into*

- $\frac{13}{14} \left( 9^{n-1} + \frac{1}{13} 2^{n-1} - \frac{7}{13} x_{n-1} \right)$  of type 01,
- $\frac{1}{14} (9^{n-1} - 2^{n-1} - 7x_{n-1})$  of type 10,
- $x_{n-1}$  of type 11.

*Proof.* This is [9, Proposition 6]. □

Tasks of types 01 or 10 have unique optimal solutions. It is interesting to note that despite their relative rarity, tasks with non-unique, namely two, minimal solutions occur, as mentioned in the introduction, for all but the perfect states.

*Remark.* The atomic structure of benzenoid hydrocarbons contains Kekulé rings (hexagons of carbon atoms), which leads to a number of mesomeres for the same structure, depending on the distribution of double bonds arranged appropriately to satisfy the tetravalency of carbon. For a specific class of condensed benzenoid hydrocarbons, where neighboring hexagons share a pair of carbon atoms, with the molecular formula  $C_{12n+2}H_{6n+4}$ , obtained by condensing  $n$  Phenanthrene structures to get a zigzag chain of  $2n$  hexagons with one extra hexagon condensed to every even numbered hexagon of the chain, the number  $K_n$  of Kekulé structures fulfills the recurrence  $K_{n+2} = 5K_{n+1} - 2K_n$  for  $n \in \mathbb{N}_0$  with  $K_0 = 1$  (Ethylene with no hexagon) and  $K_1 = 5$  (Phenanthrene with 3 hexagons) (see [5, pp. 75–78]). So  $K_n = x_{n+1}$ . Although it is not too surprising that the same recurrence might occur in different settings, this is an example of the Tower of Hanoi structure to be omnipresent.

For  $p = 4$  we need  $n = 4$  discs for an example where the bound in Theorem 1.1 is sharp, namely a task whose bitcode is larger than 3: let  $s = 0233$  and  $t = 3001$ , then the shortest paths using 1, 2 or 3 LDMs are unique, respectively, and all have length 6; the LDM code is therefore  $111 \cong 7$ .

For the even more ambitious goal to find a task where 3 LDMs are *necessary* in a shortest path on 4 pegs, we have to employ  $n = 6$  discs: the task  $022333 \rightarrow 300101$  is of type  $100 \cong 4$ ; its shortest solution is unique and contains 3 LDMs: 022333, 122333, 121333, 101333, 100333, 200333, 200332, 200302, 200102, 200101, 300101.

In fact, we have the following:

**Proposition 1.2.** *To solve the task  $022333 \rightarrow 300101$  in  $H_4^6$  employing precisely  $k \in [3]$  LDMs,  $13 - k$  moves are necessary and sufficient.*

*Proof.* Sufficiency follows by constructing appropriate solutions as the one given above for  $k = 3$ . As found out by computer search, the optimal path is unique, so that no solution using less than 3 LDMs can be shorter than 11 moves. Therefore, it suffices to prove that no solution with precisely 1 LDM exists making 11 moves only.

Let us assume in the contrary that there *is* such a solution. Then, since disc 6 necessarily moves once only from peg 0 to peg 3, there must be an  $s \in [2]^5$  such that, with

$r := 22333$  and  $t := 00101$ , we have  $d(r, s) + d(s, t) = 10$  in  $H_4^5$ , where  $d$  is the canonical graph distance, i.e. the length of a shortest path. As any solution for the task  $r \rightarrow s$  would distribute the 3-tower on peg 3 in  $r$  to the pegs 1 and 2 in  $s$  and similarly for the tower consisting of 3 discs on peg 0 in  $t$ , we know that  $x := d(r, s) \in \{4, 5, 6\}$ , because at least 4 moves are needed to transfer 3 discs from a single peg to two different ones.

One can then show by combinatorial arguments or computer search that each of the three values for  $x$  leads to a contradiction.  $\square$

It is now natural to ask whether for all  $p$  there are shortest paths with  $p - 1$  LDMS, i.e. tasks with bitcodes greater than or equal to  $2^{p-2}$ , and if so, for which numbers of discs  $n$ ; as we have seen, for  $p = 3$  this is so for all  $n \geq 2$ . The other question is, if for any given  $p$  there exists an  $n$  such that for  $H_p^n$  there are tasks with bitcode  $10\dots 0 \cong 2^{p-2}$ , i.e. which need  $p - 1$  LDMS in a shortest path. This is true for  $p = 3$  and  $n \geq 3$ . For  $p > 3$  we will first summarize some theoretical results and then approach these questions by computer experiments.

## 2 Theoretical results

To facilitate the formulation of some statements, we introduce the following notion: let  $\lambda : (\mathbb{N} \setminus \{1, 2\}) \times \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$  be defined by  $\lambda(p, n, b) = 1$ , if there exists a task with code  $b$  in  $H_p^n$ , and  $\lambda(p, n, b) = 0$  otherwise. For instance, it follows from Theorem 1.1 that  $\lambda(p, n, b) = 0$ , if  $b \geq 2^{p-1}$ . The simplest cases are the following.

**Proposition 2.1.**  $\lambda(p, n, 1) = 1$ ;  $\lambda(p, 1, b) = 0$ , iff  $b > 1$ ;  $\lambda(p, n, 3) = 1$ , iff  $n \geq 2$ ;  $\lambda(3, n, 2) = 1$ , iff  $n \geq 3$ ; for  $p \geq 4$ :  $\lambda(p, n, 2) = 1$ , iff  $n \geq 4$ .

*Proof.* For the first identity consider the task  $0r \rightarrow (p - 1)r$  with  $r \in [p - 2]^{n-1}$ , where the largest disc moves once and only once. The second statement is clear because a single disc can and will only move once in an optimal solution. For  $b = 3$  the case  $n = 1$  is already covered, and for  $p = 3$  we apply Proposition 1.1; otherwise consider the task  $0r(p - 1) \rightarrow (p - 1)r0$  with  $r \in [p - 3]^{n-2}$ , which can and has to be solved in 3 moves with either 1 or 2 LDMS. Finally, for  $b = 2$ , the case  $p = 3$  can again be seen from Proposition 1.1 and a combinatorial analysis similar to the previous one, but somewhat more involved, can be applied for the case  $p \geq 4$ ; see [1, Satz 5.6] for details.  $\square$

The general strategy used above to prove that  $\lambda = 1$  is the following: choose (computer assisted) some task with LDM code  $b$  and show that the code remains unchanged after increasing the numbers of pegs and discs. It can also be employed for  $b \in [7] \setminus [3]$ , at least if  $p \geq 5$ .

**Proposition 2.2.** Let  $p \geq 5$ . Then  $\lambda(p, n, 7) = 1$  for  $n \geq p$ ,  $\lambda(p, n, 6) = 1$  for  $n \geq p + 1$ , and  $\lambda(p, n, 5) = 1 = \lambda(p, n, 4)$  for  $n \geq p + 2$ .



(1) On the number of discs:

$$n_p = 3 + \sum_{q=4}^p (2q - 3) = 3 + \sum_{q=1}^{p-3} (2q + 3) = p(p - 2).$$

(2) Some definitions: The discs  $n_q + 1$  and  $n_q + q$ , which are on  $q$  in the starting state,  $3 \leq q \leq p - 1$ , as well as disc 2 have to be moved at least once, since their destination is on a different peg. So we call these discs *one-move-discs*. All other discs (apart from 1) must even be moved twice, as there is a larger disc below them in  $s$  which is also below them in  $t$ . They are therefore called *two-move-discs*. Moreover, every further move of these discs as well as every move of disc 1 will be called an *extra-move (EM)*. More specifically, extra-moves are all moves but the last of a one-move-disc, all but the first and the last of a two-move-disc and all moves of the largest disc.

Moreover, assume that we are in a state  $u = u_1 \dots u_n$ . We say that disc  $d$  is in *goal position* if disc  $d$  and exactly all larger discs also lying on  $u_d$  lie on the same peg in  $t$ , i.e. if for all  $j \in [d]$  it is true that  $u_j = t_d$  if and only if  $t_j = t_d$ .

(3) We now claim: *There exists a path from  $s$  to  $t$  ending with an EM from  $p - 2$  to  $p - 1$  and containing altogether  $p - 1$  EMs. Moreover, all EMs are LDMs.* We prove this by induction on  $p$ . For  $p = 3$ , the task is  $022 \rightarrow 200$ , which is solvable in five moves, the first and the last of which are LDMs and therefore EMs. Assume the claim to be true for  $p - 1$ . For the task with  $p$  pegs, first follow the path given by the induction hypothesis up to and excluding the last move, i.e. from  $s = s_1 \dots s_{n_{p-1}} s_{n_{p-1}+1} \dots s_{n_p}$  to  $(p-2)t_2 \dots t_{n_{p-1}} s_{n_{p-1}+1} \dots s_{n_p}$ , using  $p - 2$  EMs of the largest disc. The “new” discs remain untouched on peg  $p - 1$ . Note that in the definition of the task with  $p$  pegs all “old” discs (beside the largest) have the same start and goal positions as in the task with  $p - 1$  pegs. Then peg  $p - 3$  is empty. Now distribute the upper  $p - 2$  discs (which are all two-move-discs)  $n_{p-1} + p$  to  $n_{p-1} + 2p - 3$  on the  $p - 2$  pegs 0 to  $p - 4$  and  $p - 2$ . After that, one moves the one-move-disc  $n_{p-1} + p - 1$  from  $p - 1$  to  $p - 3$  and collects the distributed discs  $n_{p-1} + p$  to  $n_{p-1} + 2p - 3$  in goal position on peg  $p - 3$ . Altogether, the tower  $[B_{p-1}]$  is moved from  $p - 1$  to  $p - 3$ . Proceed analogously with the remaining discs on  $p - 1$ , the tower  $[A_{p-1}]$ , which has to be moved onto  $p - 4$ . First, distribute the upper  $p - 3$  discs, namely  $n_{p-1} + 2$  to  $n_{p-1} + p - 2$ , on the  $p - 3$  pegs 0 to  $p - 5$  and  $p - 3$ , then move  $n_{p-1} + 1$  from  $p - 1$  to  $p - 4$  and finally move the distributed discs to their goal position on  $p - 4$ . While doing this, both one-move-discs are moved exactly once, and the two-move-disc exactly twice, i.e. no extra-moves are executed. Finally, move the largest disc from  $p - 2$  to  $p - 1$ ; this is the additional extra-move.

(4) Optimality: *If the largest disc is moved differently than described in (3), more extra-moves than in (3) are needed.* To show this, let a path  $P$  from  $s$  to  $t$  be given. With respect to this path, let  $q$  be maximal with the following properties:

- (i) the largest disc 1 is the only one on peg  $q$ ,
- (ii) peg  $q + 1$  is empty,
- (iii) the pegs  $\geq q + 2$  are still in their start state,

- (iv) the pegs  $< q$  are already in their goal state, with the exception of the tower  $[A_{q+2}]$ ,
- (v) exactly  $q$  extra-moves have been executed.

In other words, up to peg  $q$ , the path  $P$  is as the one described in (3). Such a  $q$  exists because the conditions are fulfilled in the start state (with  $q = 0$ ). If  $q + 1 = p - 1$ , then the task is solved with one move of the largest disc on  $p - 1$  and  $P$  is like the path of (3). In all other cases, there is a peg still in its initial state.

Let  $r \neq q$  become the (from now on) first peg which will be empty on  $P$ . Such a peg exists, since every peg in starting state must be empty after some steps. Assume that  $r < q$ . Then  $P$  is certainly not a shortest path, as all discs on the pegs  $< q$  are already in goal position and larger than the discs still to move. Therefore, they do not obstruct them. This is not the case for the largest disc 1, but if it is moved on a peg  $< q$ , the path  $P$  is not a shortest one by the rule ‘‘They never come back’’. As  $q + 1$  is empty and  $r \neq q$  by definition, we may assume  $r \geq q + 2$ .

We may further assume that the pegs  $> r$  and the discs on them are not used until  $r$  is empty. Indeed, these discs are not used since they are smaller than the others (by the choice of  $q$ ) and therefore they would only obstruct the other discs by blocking pegs; this can also be realized by simply not using them. Moreover, since the pegs  $> r$  will not be empty before  $r$  is (by the choice of  $r$ ), these pegs cannot be used by discs from pegs  $\leq r$ .

In the sequel, we estimate the number of extra-moves that are necessary to clear peg  $r$  or that will be necessary to move the discs in their goal positions. We show by case-by-case analysis, that this number is larger than the corresponding number of the path in (3). This means that in order to move the largest disc from  $q$  to  $r$ , more EMs are needed than in the path of (3). Hence the largest disc has indeed to be moved as in (3) and therefore,  $p - 1$  LDMs are necessary.

*Case 1.*  $r = 2$  and  $q = 0$ . Thus we are in state 022. The notation  $k \xrightarrow[d]{l}$  means, that disc  $d$  is moved from peg  $k$  to peg  $l$ . The label *EM* below the arrow indicates that this move is an extra-move. With this notation, the path in (3), which contains one extra-move, is:

$$(a) \quad 0 \xrightarrow[EM]{1} 1, \quad (b) \quad 2 \xrightarrow{3} 1, \quad (c) \quad 2 \xrightarrow{2} 0.$$

Then peg  $r$  is empty. Assume first that  $P$  differs already at move (a). Then one moves either  $2 \xrightarrow{3} 1$ , followed by  $2 \xrightarrow[EM]{2} 0$ , which is an extra-move, since the one-move-disc 2 is not moved in goal position. Later, one more extra-move of 2 will be necessary. Or one moves  $2 \xrightarrow{3} 0$ , followed by  $2 \xrightarrow[EM]{2} 1$ . Then disc 3 must be moved once more, without reaching its goal position (EM). So assume now that  $P$  differs at (b). Then one extra-move, namely (a), is already executed; and the move  $2 \xrightarrow{3} 0$  follows. Therefore, disc 3 must be moved away from 0 with one extra-move. There is no possible variation at (c). In any case, more extra-moves are necessary.

Case 2.  $r = q + 2$ ,  $q \neq 0$ , i.e.

$$\frac{\text{xxxx}}{q \text{ pegs}} \quad \frac{1}{q} \quad \frac{\quad}{q+1} \quad \frac{\begin{matrix} [B_{q+2}] \\ [A_{q+2}] \end{matrix}}{q+2} \quad \cdots \quad .$$

Recall that  $[A_{q+2}]$ , or  $[A]$  for short, denotes the lower tower of the  $q + 1$  discs  $n_{q+2} + 1$  to  $n_{q+2} + q + 1$ , and  $[B_{q+2}] =: [B]$  the upper tower of the  $q + 2$  discs  $n_{q+2} + q + 2$  to  $n_{q+3}$  on  $q + 2$ ; moreover, xxxx stands for the goal distribution of discs 2 to  $n_{q+2}$ . The path in (3) is  $q \xrightarrow[EM]{1} q + 1$ ,  $q + 2 \xrightarrow{[B]} q$  and  $q + 2 \xrightarrow{[A]}$  somewhere. A variation is not essential if some discs of  $[B]$  are moved before 1 is moved on  $q + 1$ . But if all  $q + 2$  discs of  $[B]$  are moved before 1, then the one-move-disc  $n_{q+2} + q + 2$  does not reach its goal position, since 1 is on its goal peg (EM). In order to move the  $q + 1$  discs of  $[A]$  away from  $r$ , at least one peg must not be occupied by discs from  $[B]$ . Then a disc of  $[B]$  must be moved again without reaching its goal position (EM).

Case 3.  $r = q + 3$ . This means, in obvious notation,

$$\frac{\text{xxxx}}{q \text{ pegs}} \quad \frac{1}{q} \quad \frac{\quad}{q+1} \quad \frac{\begin{matrix} [B] \\ [A] \end{matrix}}{q+2} \quad \frac{\begin{matrix} [B'] \\ [A'] \end{matrix}}{q+3} \quad \cdots \quad .$$

Note that in the case  $q = 0$ ,  $[A]$  is empty. The path of (3) needs 2 extra-moves until  $q + 3$  is cleared. By definition of  $r$ , the peg  $q + 2$  was not cleared before. Thus disc 1 is on  $q$  or on  $q + 1$ .

One case is that 1 is on  $q + 1$ . To clear peg  $q + 3$ , the  $q + 3$   $[B']$ -discs must first be distributed on  $q + 3$  pegs. In doing so, the bottom-most disc of  $[B']$  does not reach its goal position, since 1 is there (1 EM). In order to move the at least two  $[A']$ -discs away, discs of  $[B']$  must not be on at least two pegs since they are smaller. Therefore two discs of  $[B']$  must be moved again without reaching their goal position (2 EMs).

The other case is that 1 is on  $q$ . Let us assume that the  $[B']$ -discs are already away from  $r = q + 3$ , possibly in their goal positions. Then the  $q + 2$   $[A']$ -discs must be distributed on  $q + 2$  pegs, because at least one peg is blocked by  $[B']$ -discs. Thereby, the lowest  $[A']$ -disc does not reach its goal position, because disc 1 is there (1 EM). If no  $[A']$ -disc is on peg  $q$ , then there is one peg less available for the distribution, i.e. one of the  $[A']$ -discs must be moved again. If some discs reached peg  $q$ , they must be moved away again (because disc 1 is there) without getting into their goal position (in both cases at least 1 EM). Furthermore, if no disc of  $[A']$  is moved onto  $q + 2$ , once again one peg can not be used for the distributions. Let us consider now the other case, i.e. some  $[A']$ -discs are moved onto  $q + 2$ . If there are still  $[B]$ -discs on  $q + 2$ , the  $[A']$ -discs must be moved away from  $q + 2$ , since they are on top of the  $[B]$ -discs on  $q$  in the goal state. If that is not the case, the lowest disc of  $[B]$  was already moved without reaching the goal position (in all subcases at least 1 EM). Note that the last subcase is impossible for  $q = 0$ , since  $q + 2$  is not cleared, but  $[A]$  is empty.

Case 4.  $r \geq q + 4$ . Thus we have the distribution

$$\begin{array}{cccccccc} \text{xxxx} & 1 & & [B] & \cdots & [B'] & [B''] & \\ \hline q \text{ pegs} & q & q+1 & q+2 & \cdots & r-1 & r & \cdots \end{array}$$

The path in (3) needs  $r - q - 1$  extra-moves. In order to clear peg  $r$ ,  $2r - 1$  discs must be distributed on  $r$  pegs. Hence  $r - 1$  discs must be moved a second time before  $n_r + 1$  is moved away from  $r$ . Thereby, none of the  $r - 1$  discs which moved twice reaches the goal position: neither do discs of  $[B'']$ , since peg  $r - 2$  is occupied by the choice of  $r$ , nor the discs of  $[A'']$ , since  $n_r + 1$  is not in goal position. This produces  $r - 1$  EMs. The lowest  $[B'']$ -disc, i.e. the one-move-disc  $n_r + r$ , does also not reach its goal position with its first move (1 EM). Altogether, at least  $r > r - q - 1$  EMs are needed.

(5) Finally we generalize this task to more discs. The case  $p = 3$  follows from Proposition 1.1. Thus we may assume  $p \geq 4$ . Then the tower  $[A_{p-1}]$  may be arbitrarily enlarged without changing the above proof essentially. The additional moves that are necessary since the tower  $[A_{p-1}]$  is larger are called *additional-moves (AMs)*. Its lowest disc remains a one-move-disc by Theorem 2.4 below. Therefore all statements on EMs above are still valid. The number of AMs is minimized, if the number of pegs available for moving the tower  $[A_{p-1}]$  is maximized. This is the case in the path considered in (3), since all smaller discs, namely the discs of  $[B_{p-1}]$ , are on exactly one peg. This means, that the AMs are independent from the EMs, i.e. no AMs can be saved by making more EMs or vice versa.  $\square$

**Remark 2.1.** In  $H_p^{p(p-2)}$  we need  $(2p - 1)(p - 2)$  moves to solve the task considered in the preceding proof. To obtain this, the number of one-move-discs is subtracted from twice the number of one- and two-move-discs and finally the number of EMs is added:

$$2(p(p - 2) - 1) - (2(p - 3) + 1) + (p - 1) = 2p^2 - 5p + 2 = (2p - 1)(p - 2).$$

**Corollary 2.1.** If  $p \geq 3$  and  $n \geq p(p - 2) - 1$ , then there exists a task on  $H_p^n$  with LDM pattern  $110 \dots 0$ , i.e.  $\lambda(p, n, 3 \cdot 2^{p-3}) = 1$ .

*Proof.* The case  $p = 3$  is already covered by Proposition 2.1. For  $p \geq 4$ , such a task can be constructed from the task of Theorem 2.1 by omitting disc 3 (we remind the reader that 1 is the largest disc). Then it is possible to move in the very beginning either  $0 \xrightarrow{\text{LDM}} 1$ ,  $2 \xrightarrow{2} 0$  and  $1 \xrightarrow{\text{LDM}} 2$  or  $2 \xrightarrow{2} 1$ ,  $0 \xrightarrow{\text{LDM}} 2$  and  $1 \xrightarrow{2} 2$ . Thus the largest disc is moved either once or twice. After that, we conclude as in the proof of Theorem 2.1. In particular, Case 3 for  $q = 0$  is also covered. Therefore  $p - 3$  more largest disc moves are necessary.  $\square$

Note that from here on we will return to the original notation for the disc labels, namely with the smallest disc denoted by 1 etc.

We now present some asymptotic results, i.e. for large numbers of discs.

**Lemma 2.1.** *If a task with LDM pattern  $b$  exists on  $H_p^n$ , then for every  $q > p$  there exists a task with LDM pattern  $b$  on  $H_q^m$  for sufficiently large  $m$ . More specifically,*

$$\forall q > p \forall m \geq 2^{n-1}(q-p) : \lambda(q, m, b) \geq \lambda(p, n, b).$$

*Proof.* Let  $\alpha \in (H_p^n)^2$  be a task with pattern  $b$ . For  $q > p$  let  $m \geq 2^{n-1}(q-p)$ . A task  $\alpha'$  on  $H_q^m$  is constructed in the following way. Pegs 0 through  $p-1$  are occupied as in  $\alpha$ , but using the  $n$  largest discs. On each of the pegs  $p$  to  $q-1$  we now place a tower of  $2^{n-1}$  discs in both, initial and target configurations of  $\alpha'$ . The remaining discs can be distributed arbitrarily on these pegs, but again identical in start and goal. Then  $\alpha'$  is also of type  $b$  because each shortest solution for  $\alpha$  is also one of  $\alpha'$  and no new shortest paths show up. The latter is obvious if pegs  $p$  to  $q-1$  are not used; if one of them is, then it must be cleared of discs and filled again which takes at least twice as many moves as there are discs on that peg, i.e. at least  $2^n$ , a number which exceeds the diameter of  $H_p^n$  (see [12, Proposition 5.36]), such that a longer path would emerge.  $\square$

**Corollary 2.2.** *Let  $p \geq 3$ . If  $q < p$ , then for all sufficiently large  $n$  there are geodesics in  $H_p^n$  containing  $q$  LDMs. For  $q \geq p$  no such geodesics exist.*

*Proof.* The statement is clearly true for  $q = 0$  and  $q = 1$ ; for  $q = p-1$  it follows from Theorem 2.1. For  $1 < q < p-1$  we first apply Theorem 2.1 for  $H_q^{n_q}$  and then use Lemma 2.1 to obtain a geodesic in  $H_p^n$  containing  $q$  LDMs, where  $n$  has to be large enough. For  $q \geq p$ , Theorem 1.1 excludes the possibility of a geodesic with  $q$  LDMs.  $\square$

In special cases the maximum number of LDMs is smaller. For instance, we have

**Theorem 2.2.** *If  $n < p$ , i.e. if there are more pegs than discs, the largest disc moves at most twice; in other words,  $\lambda(p, n, b) = 0$ , if  $b \geq 4$ .*

*Proof.* This is a special case of the subsequent Theorem 2.3.  $\square$

**Theorem 2.3.** *Let  $n < p$ . Then every disc is moved at most twice on a shortest path between any two vertices of  $H_p^n$ .*

Note that the inequalities in the two preceding theorems are sharp. For the proof of Theorem 2.3 we need the following lemma.

**Lemma 2.2.** *Let  $n < p$  and let two states  $s$  and  $t$  be given. Then either there is a peg which is empty in  $s$  and also in  $t$ , or it is possible to move some disc with one move directly to its position on the bottom of some peg in state  $t$ .*

*Proof.* Let  $o$  be the number of occupied pegs and  $e$  the number of empty pegs in the actual state  $s$ , i.e.  $o + e = p$ . Suppose both alternatives do not hold. Then all  $e$  pegs, which are now empty, are occupied in the goal state. Moreover, the  $e$  discs, which will be at the bottom of these  $e$  pegs, are not on the top of the  $o$  occupied pegs. Thus the  $o$  discs, which are on top of the occupied pegs, are different to the former  $e$  discs. Hence there are at least  $o + e = p > n$  discs, a contradiction.  $\square$

*Proof of Theorem 2.3.* Let  $P$  be a path from  $s$  to  $t$  in  $H_p^n$ . Let us call the last move of a disc in this path its *goal-move* and all other moves *avoiding-moves*. Assume that a disc  $d$  is moved three times:  $e_1 : i_1 \rightarrow i_2$ ,  $e_2 : i_2 \rightarrow i_3$  and finally  $e_3 : i_3 \rightarrow i_4$ ; thus  $e_3$  is the goal-move of  $d$ . Note that the involved pegs  $i_k \in [p]_0$  are not necessarily different. We now describe an algorithm that sets out from  $P$  and constructs a shorter path from  $s$  to  $t$ , in which disc  $d$  is moved at most twice. We follow  $P$  until  $e_1$ . Just before  $e_1$ , we execute goal-moves until  $d$  is in its goal position or there is an empty peg which is also empty in the goal state. Since there are at most  $n$  goal-moves, we are in such a state after finitely many moves by Lemma 2.2. These goal-moves are not additional moves, but just executed earlier. If  $d$  is not yet in its goal position, we move disc  $d$ , instead of following  $e_1$ , from  $i_1$  to  $j$ , where  $j \in [p]_0$  is a peg empty now and also in the goal state. Then we follow  $P$  until  $e_3$ , but omit the move represented by  $e_2$  and all those goal-moves which have already been executed. Instead of the now occupied peg  $j$ , on which discs are placed in  $P$  with avoiding-moves only, we use another peg empty now and in the goal state if the avoiding-move is still necessary. Such a peg exists by Lemma 2.2 after some goal-moves executed in advance. If necessary, we iterate this procedure several but finitely many times. If the goal-move of  $d$  has not already been executed, we move it from  $j$  into goal position instead of using  $e_3$ . Finally we follow  $P$  until  $t$ , but omit, of course, already executed goal-moves. Altogether, we saved at least one move, namely  $e_2$ , and the theorem follows.  $\square$

The reader is invited to sketch the algorithm, e.g., with the example of the path 000, 003, 023, 021, 031, 231, 201 in  $H_4^3$ , containing three moves of disc 2. Note that we did *not* claim that the new path is optimal; this is not true, neither in the example, nor in general. But it is shorter than the original one and employs at most two moves of disc  $d$ .

A direct consequence of Theorem 2.3 is that  $\text{diam}(H_p^n) \leq 2n$  for  $n < p$ . Again based on Lemma 2.2, this upper bound can be improved to read  $\text{diam}(H_p^n) \leq 2n - 1$  (see [12, Proposition 5.38]). On the other hand it is clear that the perfect-to-perfect task necessarily needs  $2n - 1$  moves ( $n - 1$  discs have to move before the (first) move of disc  $n$  and again after the (last) move of  $n$ ) such that we arrive at

**Proposition 2.3.** *For  $n < p$ , we have  $\text{diam}(H_p^n) = 2n - 1$ .*

The strongest result on LDMs holds, of course, when perfect states are involved (see [20, Lemma 4] and, rather implicitly, [22, Theorem 2.2]).

**Theorem 2.4.** *If the initial or the goal state is perfect, then the largest disc moves at most once in a shortest path.*

We base our proof on the following observation.

**Lemma 2.3.** *If the two pegs  $k$  and  $l$  are empty in some state  $t$  of a shortest path between any two states of  $H_p^n$ , then the path does not contain an LDM from  $k$  to  $l$ .*

*Proof.* Let us suppose that there is an LDM between pegs  $k$  and  $l$ . Since the graph is undirected, taking  $t$  as the target of a sub-path containing this move is no loss of generality. Since both pegs are empty in  $t$ , it also makes no difference whether we consider a  $ks, t$ -path (via  $ls$ ) or an  $ls, t$ -path (via  $ks$ ), where  $s \in [p]_0^{n-1}$  with  $k \neq s_i \neq l$  for all  $i \in [n-1]$ . So we may assume that the LDM from peg  $k$  to peg  $l$  is the first move on a shortest path from some  $ks_{n-1} \dots s_1$  to  $t = t_n \dots t_1$  with  $k \neq t_i \neq l$  for all  $i \in [n]$ . Consider the sub-path from  $ls_{n-1} \dots s_1$  to  $t$  and swap the roles of  $k$  and  $l$  (cf. [22, p. 17]). Then we get a path from  $ks$  to  $t$  which is shorter by one step than the original one, leading to a contradiction.  $\square$

*Proof of Theorem 2.4.* Assume that the goal is perfect on peg  $j$ . (If the initial state is perfect, we consider the reverse path.) Because all pegs but  $j$  are empty in the goal state, the largest disc does not move from  $k$  to  $l$  for all  $k \neq j \neq l$  according to Lemma 2.3. Since the largest disc will never return to a peg it has left (cf. the proof of Theorem 1.1), it will never be moved away from  $j$ . Therefore there exists at most one move of the largest disc (namely onto  $j$ ).  $\square$

### 3 Numerical investigations

We are not only interested in upper bounds on the number of LDMs, but also in the existence of a task with a given LDM pattern. For example, are there tasks in  $H_p^n$  whose shortest path may have 2 or 5, but not 1, 3, 4 or more than 5 LDMs, i.e., in binary expression 10010. What are the minimum numbers of pegs and discs for such a pattern of LDMs to occur? Unfortunately, we have only few analytical methods to construct such tasks or to prove their (non-)existence; they are employed in the proofs of the above theorems. Instead we computed LDMs numerically. For moderate numbers of pegs and discs it is possible to search in the Hanoi graphs for shortest paths by standard methods. To analyze LDMs, however, we used a modified algorithm, which spans a layered data structure over the graph in a breadth-first search fashion.

#### 3.1 The program

The goal of our program is to assign the bit vector  $b$  to each vertex (cf. Section 1). In ordinary breadth-first search algorithms as applied for a connected graph (cf., e.g.,

[4, Section 22.2]) the root vertex is first put into level 0 and the algorithm finds all its neighbors and puts them into level 1. Then the algorithm chooses all unvisited neighbors of vertices in level 1 and puts them into level 2, etc. A vertex is marked as visited (and its predecessor is recorded) immediately after it is put into its level, so that it cannot be visited twice. The generating of levels continues until every vertex of the graph has been reached. At the end of this process a layered data structure is obtained from which it is straightforward to deduce the distance and exactly one shortest path between the root vertex and any other vertex of the graph.

In the modified algorithm, instead of immediately marking a vertex as visited, we update the bit vector accompanying the vertex as many times as there are adjacent vertices on the previous level. If the largest disc is moved in the transition from vertex  $u$  to vertex  $v$ , then the bit vector of  $v$  is updated as follows (all operations bitwise; “ $\ll 1$ ” means left-shift of bit vector):

```

if (LDMs(u)==0) {
  LDMs(v) := LDMs(v) OR 1;
} else {
  LDMs(v) := LDMs(v) OR (LDMs(u) << 1);
}

```

otherwise, if a smaller disc is moved, we update the bit vector of  $v$  as follows

```

LDMs(v) := LDMs(v) OR LDMs(u).

```

Only after generating the next level is accomplished, all its vertices are marked as visited and their bit vectors will not change anymore. They now contain the information about the LDMs from the root to these vertices.

### 3.2 Search for tasks with LDM code $2^{p-2}$

Is the minimal disc number for a task of type  $10\dots 0 \cong 2^{p-2}$  in Theorem 2.1 sharp? For  $p = 3$  the answer is “yes” as can be seen from Proposition 1.1. For  $n = 4$ , however, the example of the task  $022333 \rightarrow 300101$ , which has a unique solution using 3 LDMs, shows that  $n = 6$  discs are enough instead of the 8 predicted by the theorem. And, incredible as it might sound, the task  $033324240 \rightarrow 411001202$  in  $H_5^9$  can only be solved making 4 moves of disc 9. So it seems more likely that  $3(p - 2)$  would be the better bound in Theorem 2.1! (That fewer discs will not produce tasks of type  $2^{p-2}$  for  $p = 4$  and  $p = 5$  has been checked numerically; cf. Section 3.3.)

To test this conjecture for  $p = 6$ , i.e. to decide whether the bound is 12 or not in that case, we have to consider  $H_6^n$  up to  $n = 12$ . Another educated guess, namely that there exist tasks of type  $2^{p-1} - 1$ , i.e. solutions making  $p - 1$  LDMs, as soon as  $n \geq 2(p - 2)$ , has been verified by our experiments (see below) up to and for  $p = 6$ , in particular in  $H_6^8$ .

However, the huge extent of  $H_6^{12}$ , which has more than 2 billion<sup>2</sup> vertices and more than 16 billion edges, will not allow for a complete search in this case. On the other hand, to prove the existence in  $H_6^{12}$  of a task with LDM type 16, a complete search, comprising *all* vertices as roots, is not necessary; it suffices just to find *one* such task. Therefore, if we apply the algorithm described in Section 3.1 to some “guessed” task, we may verify that it indeed *is* a proper candidate. But which vertex to choose for the root of the layered structure to be constructed? A first few clever guesses turned out to be not so clever after all. However, this did not mean that we could not do better! Even if we reduce the number of initial vertices making use of the symmetries, we still have  $6^{12}/6!$ , i.e. more than 3 million left, which did not allow us to carry out our algorithm on all of them. Fortunately, it is possible to reduce the number of candidates for an initial vertex further by analytical considerations. For this, we say that a vertex has *property*  $\ll A \gg$ , if the largest disc is alone on its peg and some other peg is empty, i.e. if the largest disc can move immediately.

**Lemma 3.1.** *If there is a task in  $H_p^n$  with LDM code  $2^{p-2}$ , then there exists a task with code  $2^{p-2}$  whose initial state has property  $\ll A \gg$ .*

*Proof.* Take an optimal solution for the given task and replace its initial state by the state immediately preceding the first move of the largest disc. The new path will be optimal being a sub-path of an optimal path.  $\square$

**Lemma 3.2.** *If a vertex  $s$  of  $H_p^n$  has property  $\ll A \gg$  and there is another empty peg, then  $s$  cannot be the initial vertex of a geodesic with  $p - 1$  LDMs.*

*Proof.* By Lemma 2.3, no shortest path contains an LDM between two pegs empty in  $s$ .  $\square$

**Lemma 3.3.** *If a vertex  $s$  of  $H_p^n$  has property  $\ll A \gg$  and there is another peg holding one disc only, then  $s$  cannot be the initial vertex of a task with LDM code  $2^{p-2}$ .*

*Proof.* The proof is by contradiction. We may assume that disc  $n$  is alone on peg 0, peg 1 is empty, and disc  $d \in [n - 1]$  is alone on peg 2. Then disc  $n$  will not move to another peg before it moves to 1, because otherwise we get a shorter solution by moving it to peg 1 in the very first move. We now distinguish the cases whether it moves from 1 to 2 directly or not.

In the former case we move disc  $d$  to peg 1 and then  $n$  to 2. Now we follow the given path, deleting all moves of disc  $n$  and swapping—similarly as in the proof of Lemma 2.3—the roles of pegs 1 and 2 up to and excluding the original move of disc  $n$  from 1 to 2. Now peg 1 is empty and peg 2 is only occupied by the largest disc, so that we arrived, with the same number of moves, at the same state as on the old path. However, instead

---

<sup>2</sup>here 1 billion means  $10^9$

of two moves of disc  $n$ , it was only moved once and the other move was compensated by one of disc  $d$ .

For the second case we move  $d$  from 2 to 1,  $n$  from 0 to 2 and then  $d$  back to 2. After these three moves we are back in a state like  $s$  but with disc  $n$  already on peg 2. Now we follow the old path avoiding all moves of disc  $n$  until it arrives on peg 2 on the old path. This saves at least three moves (of disc  $n$ ), compensating the ones made at the beginning of the new path. But the latter uses strictly fewer LDMs.  $\square$

We summarize our observations on candidates for LDM-type  $2^{p-2}$  tasks.

**Corollary 3.1.** *If there is a task in  $H_p^n$  with LDM code  $2^{p-2}$ , then there exists a task with the same code satisfying the following conditions:*

- *disc  $n$  is alone on peg 0;*
- *peg 1 is empty;*
- *the other discs occupy the remaining pegs in such a way that each of them contains at least two discs.*

Coming back to the special graph  $H_6^{12}$ , we want to determine the number of roots to be analyzed according to Corollary 3.1. The eleven smaller discs have to be distributed on the four pegs 2 to 5 with none of these pegs holding less than two discs. There are three cases.

*Case 1.* One peg is filled with two, the others with three discs each. There are as many states of that type as there are permutations of 22333444555. Taking into account the symmetries of exchanging the pegs occupied with three discs, we may divide by  $3!$ , such that we obtain for the number of roots to be considered:

$$\frac{11!}{2!3!3!3! \cdot 3!} = 15400$$

*Case 2.* Now we distribute the eleven discs such that two pegs hold two discs each and the other two pegs are occupied by three and four discs, respectively. Similarly to the above analysis this leads to

$$\frac{11!}{2!2!3!4! \cdot 2!} = 34650$$

more roots to be considered.

*Case 3.* Finally, we can have three pegs holding two discs each and one peg with five. This results in another

$$\frac{11!}{2!2!2!5! \cdot 3!} = 6930$$

vertices.

Altogether, “just”  $15400 + 34650 + 6930 = 56980$  runs of the algorithm have to be performed.

### 3.3 Numerical results

Our algorithm from Section 3.1 has been exhaustively applied to those values of  $p$  and  $n$  where the running-times were accessible. They were systematically applied to those initial states, i.e. roots, which would give us information about our conjectures and expectations, based on the theoretical insight we obtained in Sections 2 and 3.2.

The results of our computations are summarized in coded form in Figure 4 showing individual tables for each  $p$  from 3 to 7. All patterns  $b$  of LDMs which are allowed for the given number of pegs by Theorem 1.1 are represented in the left part of each table. In their right parts, headed by the number of discs  $n$ , a black field indicates that there exist tasks with the given pattern of LDMs and number of pegs and discs, i.e. that  $\lambda(p, n, b) = 1$ , whereas a white field means that this is not the case, i.e. that  $\lambda(p, n, b) = 0$ . These facts have either been obtained by our above theoretical results or by computation as described. Moreover, the tables contain black and white cases marked with a “C” indicating that these instances are conjectured by the subsequent Conjectures 1 and 2. For the remaining cases, filled in gray, the value for  $\lambda$  is neither known nor anticipated.

An obvious first expectation is:

**Conjecture 1.** *If an LDM pattern occurs in  $H_p^n$ , then it appears in all  $H_p^m$  for  $m \geq n$ , i.e.  $\lambda$  is monotone non-decreasing in  $n$ .*

A second conjecture will follow later from the quantitative results of the computations which have been carried out.

There are 8 tasks in  $H_4^4$ , where shortest paths with 3 LDMs exist, such as (0233, 3001), which we mentioned before. Tasks as, e.g., (0323, 3100) even have two shortest paths with 3 LDMs. On the other hand, all of them belong to case 111, i.e. there are also shortest paths with two and even one LDM(s). For  $p = 4$ , the case where 3 LDMs become *necessary* occurs for the first time for  $n = 6$  and there are 84 such tasks; one of these, with a unique optimal solution, has been discussed in Proposition 1.2. Some tasks, like, e.g., (011332, 300022), have as much as four shortest paths containing 3 LDMs each. It should be clear by now that such detailed information about LDM patterns can not be obtained easily for larger values of  $p$  and  $n$ .

### 3.4 Discussion

Now let us have a closer look at Figure 4. Two white entries in the table for  $p = 6$  are disturbing because they do not follow the general pattern. We read  $\lambda(6, 12, 16) = 0$ , a result which we found by employing the detailed reduction analysis from Section 3.2. So it turns out that the bound  $p(p - 2)$  from Theorem 2.1 can not be replaced simply by  $3(p - 2)$  as the cases up to and including  $p = 5$  had suggested. Moreover, these experiments revealed the values  $\lambda(6, 12, b) = 1$  for all other types  $b$ , except  $b = 17, 18, 19$ , where no conclusion could be drawn because our search was not exhaustive.

- pattern of LDMs, e.g. 4, 2, and 1, but not 3 LDMs
- no task exists
- some tasks do exist
- ◻ conjectured that no task exists
- ◼ conjectured that some tasks do exist
- trendless case

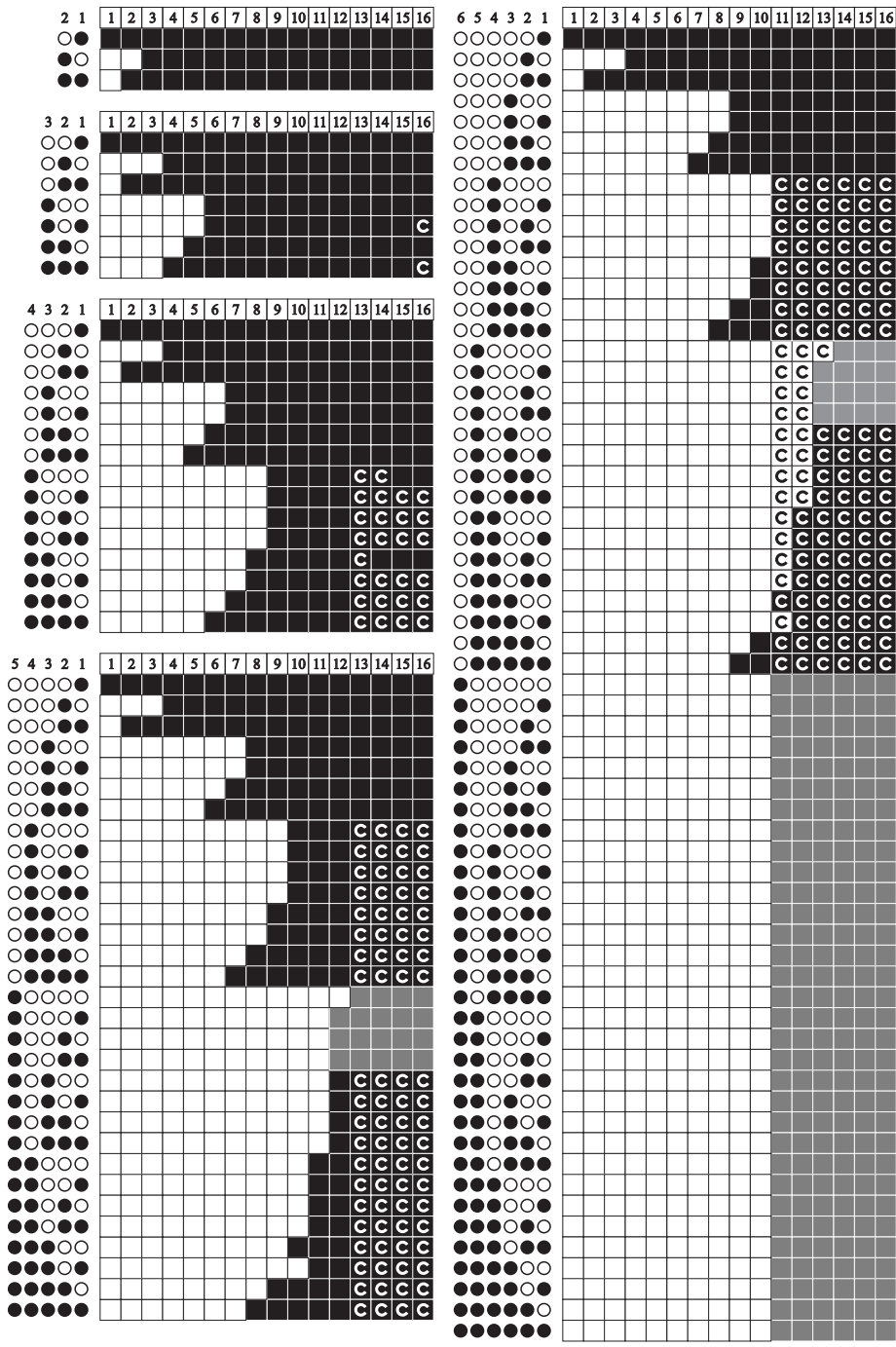


Figure 4: Patterns of LDMs

The other somewhat unexpected item for  $p = 6$  is the white case for code 11101 at  $n = 10$ , i.e.  $\lambda(6, 10, 29) = 0$ , because it destroys the regular impression one gets from the “steps” of the occurrences of black fields in the upper parts of the table. In fact, the upper halves of the tables seem to be inherited from the table before, i.e. from  $p$  to  $p + 1$  in the sense of the following conjecture.

**Conjecture 2.** *If  $3 < b < 2^{p-1}$ , then  $\lambda(p + 1, n + 1, b) = \lambda(p, n, b)$ .*

Even if we knew this conjecture to be true, it would not help us to construct all further tables recursively, because we have no idea how the lower half for  $p + 1$  is composed. This lack of insight is imposingly demonstrated by the very last assured entry, namely  $\lambda(7, 10, 63) = 0$ , which means that our other first impression that  $b = 2^{p-1} - 1$  will occur for  $n \geq 2(p - 2)$ , supported by  $p < 7$ , is not true in general.

As mentioned in the introduction, our numerical experiments were mainly motivated by the open problem to find a shortest solution of the perfect-to-perfect task for more than 3 pegs and the corresponding *Frame-Stewart Conjecture*, namely that Frame’s strategy, and Stewart’s for that matter, are in fact optimal (see [12, Chapter 5] for a comprehensive discussion). A principal feature of Frame’s algorithm is that at half-way, i.e. before the only move of the largest disc, all discs are distributed among the pegs in such a way that every peg holds a *subtower*, i.e. a subset consisting of consecutive discs. Frame’s assumption that this will lead to an optimal solution is obviously true for the (unique) shortest solution if  $n < p$ . On the other hand, the example  $p = 4 = n$  shows that optimal *non-subtower solutions* may exist, like, e.g., the one from  $0^4$  to  $3^4$  passing through 0121 (cf. [12, p. 176]). Again, numerical experiments (performed for  $p \leq 6$  and  $n \leq 14$ ; see [12, Table 5.5]) give rise to a conjecture, namely

**Conjecture 3.** *For  $n \geq \binom{p}{2}$  there are no optimal non-subtower solutions for the perfect-to-perfect task in  $H_p^n$ .*

Together with the result that the Frame-Stewart Conjecture is in fact true for  $n \leq \binom{p}{2}$  [12, Proposition 5.20], this, even if proved only for  $n > \binom{p}{2}$ , would justify Frame’s assumption.

It seems that a lot more insight into the structure of Hanoi graphs  $H_p^n$  is necessary to decide on questions of metric properties of these graphs. This becomes obvious by considering a variant of the Tower of Hanoi, the *Switching Tower of Hanoi*, which had been introduced by S. Klavžar and U. Milutinović in [17]. It led to the state graphs called *Sierpiński graphs*  $S_p^n$ . Isomorphic to  $H_p^n$  for  $p = 3$ , they have a much simpler structure than their counterparts for larger  $p$ , because they are essentially iterated complete graphs. It turns out that there are at most 2 LDMS for any Sierpiński task [1, Satz 7.2] (also implicit in the proof of [17, Theorem 5]) and that the only possible non-zero codes  $b \in [3]$  occur in  $S_p^n$  for the same values of  $n$  as in  $S_3^n \cong H_3^n$  [1, Satz 7.3]. One can even obtain explicit formulas as in our Proposition 1.1; see [10, Section 3.1]. Sierpiński graphs are therefore a good model to test strategies for Hanoi graphs, like, e.g., to find colorings (cf. [13]), but

it will be a long way to extend metric properties, such as eccentricities (cf. [14]) from  $S$  to  $H$  because of the complicated patterns of LDMs revealed in this study.

## Acknowledgements

We thank Andreas Groh (Munich) for supporting discussions and computations. We are also grateful to an anonymous reviewer for the comprehensive list of suggestions to make the text more comprehensible, in fact more accurate.

## References

- [1] S. Aumann, Über die Anzahl der Stege auf Geodäten von Hanoi-Graphen, Diploma thesis, LMU Munich, München, 2009.
- [2] J.-P. Bode, A. M. Hinz, Results and open problems on the Tower of Hanoi, *Congr. Numer.* **139** (1999) 113–122.
- [3] N. Claus de Siam (=É. Lucas), La Tour d’Hanoï, Jeu de calcul, *Science et Nature* **1/8** (1884) 127–128.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, Third Edition, The MIT Press, Cambridge MA, 2009.
- [5] S. J. Cyvin and I. Gutman, *Kekulé Structures in Benzenoid Hydrocarbons*, Springer, Berlin, 1988.
- [6] H. E. Dudeney, The Reve’s puzzle, The Canterbury Puzzles, *The London Magazine* **8**(no. 46) (May 1902) 367–371.
- [7] J. S. Frame, Solution to advanced problem 3918, *Amer. Math. Monthly* **48** (1941) 216–217.
- [8] K. A. M. Götz, Äquivalenzklassen von Turm-Aufgaben, Diploma thesis, LMU Munich, München, 2008.
- [9] A. M. Hinz, The Tower of Hanoi, *Enseign. Math. (2)* **35** (1989) 289–321.
- [10] A. M. Hinz, C. Holz auf der Heide, An efficient algorithm to determine all shortest paths in Sierpiński graphs, *Discrete Appl. Math.* **177** (2014) 111–120.
- [11] A. M. Hinz, S. Klavžar, U. Milutinović, D. Parisse and C. Petr, Metric properties of the Tower of Hanoi graphs, *European J. Combin.* **26** (2005) 693–708.
- [12] A. M. Hinz, S. Klavžar, U. Milutinović, and C. Petr, *The Tower of Hanoi—Myths and Maths*, Springer, Basel, 2013.
- [13] A. M. Hinz, D. Parisse, Coloring Hanoi and Sierpiński graphs, *Discrete Math.* **312** (2012) 1521–1535.
- [14] A. M. Hinz, D. Parisse, The Average Eccentricity of Sierpiński Graphs, *Graphs Combin.* **28** (2012) 671–686.

- [15] D. Klahr, Goal Formation, Planning, and Learning by Pre-School Problem Solvers or: “My Socks are in the Dryer”, in: R. S. Siegler (Ed.), *Children’s Thinking: What develops?*, Lawrence Erlbaum, Hillsdale NJ, 1978; 181–212.
- [16] D. Klahr, M. Robinson, Formal Assessment of Problem-Solving and Planning Processes in Preschool Children, *Cogn. Psychol.* **13** (1981) 113–148.
- [17] S. Klavžar, U. Milutinović, Graphs  $S(n, k)$  and a variant of the Tower of Hanoi problem, *Czechoslovak Math. J.* **47** (122) (1997) 95–104.
- [18] S. Klavžar, U. Milutinović and C. Petr, On the Frame-Stewart algorithm for the multi-peg Tower of Hanoi problem, *Discrete Appl. Math.* **120** (2002) 141–157.
- [19] X. Lu, Towers of Hanoi Graphs, *Int. J. Comput. Math.* **19** (1986) 23–38.
- [20] S. E. Park, The Group of Symmetries of the Tower of Hanoi Graph, *Amer. Math. Monthly* **117** (2010) 353–360.
- [21] D. Romik, Shortest paths in the Tower of Hanoi graph and finite automata, *SIAM J. Discrete Math.* **20** (2006) 610–622.
- [22] A. Rukhin, *On the Generalized Tower of Hanoi Problem: An Introduction to Cluster Spaces*, Master thesis, University of Maryland, College Park MD, 2004.
- [23] B. M. Stewart, Solution to advanced problem 3918, *Amer. Math. Monthly* **48** (1941) 217–219.
- [24] D. Wood, Adjudicating a Towers of Hanoi Contest, *J. Comput. Math.* **14** (1983) 199–207.