Jan Długosz University of Częstochowa
Scientific Issues, Mathematics XII, Częstochowa 2007

UNDER AND EXACT ESTIMATES OF COMPLEXITY OF ALGORITHMS FOR MULTI-PEG TOWER OF HANOI PROBLEM

Sergey Novikov

University of Podlasie, Siedlce, Poland
e-mail: novikov@poczta.onet.pl

Abstract. It is proved under and exact estimates of complexity of algorithms for the multi-peg Tower of Hanoi problem with the limited number of discs.

1. Introduction

In 1884 famous French mathematician Edouard Lucas formulated and solved a mathematical problem "The Tower of Hanoi" [1]. To solve the problem E. Lucas proposes a recurrence algorithm. The complexity of this algorithm was described with the help of the formula

$$H_3(n) = 2^n - 1, \tag{1}$$

where $H_3(n)$ is the minimum number of moves needed to solve the puzzle with n discs for three pegs.

Later different generalizations of the classical Tower of Hanoi problem were published, one of which is "The multi-peg Tower of Hanoi". The multi-peg Tower of Hanoi problem consists $k > 3$ pegs $(B_1, B_2, \ldots, B_k)$ mounted on a board together with n discs of different sizes $(1, 2, \ldots, n)$. Initially these discs are placed on one peg (B_1) in order of size, with the largest (n-disc) on the bottom. The rules of the problem allow discs to be moved one at a time from one peg to another as long as a largest disc is never placed on top of a smaller disc. The goal of the problem is to transfer all n discs to another peg (B_2) with the minimum number of moves, denoted $H_k(n)$. The function $H_k(n)$ characterizes the complexity of the algorithm for the solution to the multi-peg Tower of Hanoi problem. For an optimal solution to the k-peg version of the classic Tower of Hanoi problem also is needed recurrence algorithms.

Algorithms of moving discs from one of k pegs $k > 3$ to another peg with the help of the more than one subsidiary pegs are used today in many computer science textbooks to demonstrate how to write a recursive algorithm or program. Also these algorithms are often proposed from programming on different olympiads and competitions for informaticians. It is known the difficulty of proofs of the complexity of recursive algorithms for the multi-peg Tower of Hanoi problem. Therefore this problem is interesting for mathematicians.

The Bibliography of P.K.Stockmeyer, which is devoted to the Tower of Hanoi problem maintains more than 200 relevant positions [2] However, an optimal solution to the k-peg version of the classic Tower of Hanoi problem is unknown for each $k \geq 4$.

In [3] algorithms with upper estimates of complexity for the k-peg Tower of Hanoi problem were published (cases: $k = 4, k = 5$ for any $n \geq 2$ and the case: any k for $k \leq n \leq \frac{k*(k-1)}{2}$).

The main aim of this paper is presentation of algorithms with exact estimates of complexity for the multi-peg Tower of Hanoi problem with the limited number of discs. To prove the optimality of the algorithm $A*$ we must take out the upper estimate (non-recursive) formula of the complixity for this algorithm $H(A*, k, n)$ and to prove that any other algorithm does not allow to solve our problem better (with smaller number of moves for the same parameters k, n).

It is obvious that for any $A*$ is evenly the inequality $H(A*, k, n) \geq H_k(n)$.

If we can estimate from below function $H_k(n)$ with the help of the general argumentations we get the under estimate $H(Min, k, n)$, where $H(Min, k, n) \leq H_k(n)$.

The algorithm $A*$ allows to find the optimal solution of our problem if it is proved that $H(A*, k, n) = H(Min, k, n)$. Then we have $H(A*, k, n) = H_k(n)$ and $A*$ is the optimal algorithm.

2. An upper estimate of the function $H_k(n)$

For investigation of our problem is interesting a next algorithm for transporting n discs from the first peg to B_2 for the case of $k \geq 3$ pegs.

Algorithm A_0

1. Move $k - 2$ smallest discs from the first peg to the peg B_k.
2. Move $k - 2$ next discs from the first peg to the peg B_{k-1}.
3. Move discs from the B_k to the peg B_{k-1}.
4. Move $k - 2$ next discs from the B_1 to the peg B_k.
5. Move discs from the B_{k-1} to the peg B_k.

At last (on step $2l - 1$, where $l =]n/(k-2)[$) we move rest largest discs from the first peg to the peg B_2.

$2l$. Move all discs from the B_{k-1} (if l is not an even number) or from the B_k (if l is an even number) to the peg B_2.

In accordance with the algorithm we transport the stack of $k-2$ largest discs (from B_1 to the peg B_2) only one time. The next stack of $k-2$ largest discs we transport twice (from B_1 to B_k or B_{k-1} and then from B_k or B_{k-1} to the B_2). The next stack of $k-2$ discs we transport four times. At last the stack of $k-2$ smallest discs we transport 2^{l-1} times, where $l =]n/(k-2)[$.

We sum our moves of stacks for all steps and obtain $2^l - 1$ moves of stacks. For transporting of one stack with $k-2$ discs is needed $2k-5$ moves. Only for one stack (largest discs) less moves may be needed.

Then the explicit estimate of the comlexity of this algorithm is equal to

$$H(A_0, k, n) = (2^{]\frac{n}{k-2}[} - 1)(2k-5) \qquad (2)$$

Formula (2) for the case $k = 3$ changes into formula (1). However for cases $k \geq 4$ the solution to our problem by the algorithm A_0 is not optimal. For example in the case $k = 7, n = 21$ we have 279 moves with the help of formula (2) and we have 71 moves with the help of our formula (6) from [3].

Then formula (2) is the upper estimate of the function $H_k(n)$. The algorithm A_0 has one quality. This algorithm may be used for our problem without restrictions for parameters k and n.

3. Domains for investigations

The Tower of Hanoi (in classic version) is a well known NP problem. Obviously the problem multi-peg Tower of Hanoi is also NP problem.

Upper estimates of the complexity of all known algorithms are functions exponential (sum of power of 2). In [4] seven algorithms for multi-peg Tower of Hanoi problem were analysed. All of them have the equivalent complexity, which is a function exponential. Upper estimates for functions $H_4(n)$ and $H_5(n)$ from [3] are also exponential functions.

However for some interesting cases, described by correlation between parameters k and n, we can propose easy algorithms for the multi-peg Tower of Hanoi problem, where the calculating complexity is not an exponential function.

Let's consider next cases:

A) $n \leq C^1_{k-1}$,
B) $C^1_{k-1} < n \leq C^2_k$,
C) $C^2_k < n \leq C^3_{k+1}$,
D) $C^3_{k+1} < n \leq C^4_{k+2}$.

4. Estimations of the function $H_k(n)$ for the case A)

Theorem 1. If $k \geq 3$ and $n \leq C_{k-1}^1$, then the exact estimation of the function $H_k(n)$ is equal to

$$H_k(n) = 2n - 1. \tag{3}$$

Proof. Let's infer the under estimate $H(Min, k, n)$ of the function $H_k(n)$ for our case A).

In this case for transferring each disc (except for the largest n-disc) from the first peg to the peg B_2 in according with the rules of the problem Tower of Hanoi two moves indepently of an algorithm of transferring are needed. The largest n-disc may be transferred from the first peg to the peg B_2 with one move. We sum our moves in this case and obtain the estimation

$$H(Min, k, n) = 2(n-1) + 1 = 2n - 1$$

For getting the upper estimate of the function $H_k(n)$ we use the next simple algorithm of transferring of discs.

Algorithm A_1

[1] Move n discs from the first peg to B_2, B_3, ... , B_{n+1} so that on the each peg one disc is placed and the largest n-disc is placed on B_2.

[2] Move $n-1$ discs from temporal pegs to the peg B_2, where one n-disc is already placed.

We sum our moves and obtain

$$H(A_1, k, n) = n + n - 1 = 2n - 1.$$

Therefore, for the case A) we have $H(Min, k, n) = H(A_1, k, n)$ and our algorithm A_1 is the optimal algorithm. Then the exact estimate for the case A) is equal to $H_k(n) = 2n - 1$.

5. Estimations of the function $H_k(n)$ for the case B)

Theorem 2. If $k \geq 3$ and $C_{k-1}^1 < n \leq C_k^2$, then the exact estimation of the function $H_k(n)$ is equal to

$$H_k(n) = 4n - 2k + 1. \tag{4}$$

Proof. Let's infer the under estimate $H(Min, k, n)$ of the function $H_k(n)$ for our case B).

In this case for transferring each of some smallest discs, which are marked as n_s, (except largest $n_l = k-1$ discs) from the first peg to the peg B_2 in according with rules of the problem Tower of Hanoi four moves independly of an algorithm of transferring are needed (from B_1 to B_h, from B_h to B_i, from B_i to B_j,from B_j to B_2). All largest $k-1$ discs may be transferred from the first peg to the peg B_2 no less than with $2k-3$ moves (with the help of (3)).

We sum our moves in this case and obtain the estimation

$$H(Min, k, n) = 4(n-(k-1)) + 2k - 3 = 4n - 2k + 1.$$

For getting the upper estimate of the function $H_k(n)$ we use the next algorithm [3] of transferring of discs.

Algorithm A_2

1. Move $k-1$ smallest discs from the first peg to the peg B_k ($2k-3$ moves) by the algorithm A_1.
2. Move $k-2$ next discs from the first peg to the peg B_{k-1} ($2k-5$ moves) by the algorithm A_1.
3. Move $k-3$ next discs from the first peg to the peg B_{k-2} ($2k-7$ moves) by the algorithm A_1.

At last on stage $k-1$ we move one n-disc from B_1 to B_2.

k. Move two discs ($(n-2)$-disc and $(n-1)$-disc) from B_3 to B_2.

$k+1$. Move three next discs from B_4 to B_2.

At last on stage $2k-3$ we move rest smallest $k-1$ discs from B_k to B_2 by the algorithm A_1.

We sum our moves and obtain [3]

$$H(A_2, k, n) = 4n - 2k + 1.$$

Then

$$4n - 2k + 1 \leq H_k(n) \leq 4n - 2k + 1.$$

This yields our theorem.

6. Estimations of the function $H_k(n)$ for the case C)

We will prove the next statement

Theorem 3. If $k \geq 3$ and $C_k^2 < n \leq C_{k+1}^3$, then the exact estimation of the function $H_k(n)$ is equal to

$$H_k(n) = 8n - 2k^2 + 1. \tag{5}$$

Proof. Let's infer the under estimate $H(Min, k, n)$ of the function $H_k(n)$ for our case C).

In this case for transferring each of some smallest n_s discs from the first peg to the peg B_2 in according with rules of the problem Tower of Hanoi 2^3 moves independly of an algorithm of transferring are needed. Obviously, the number of such discs is equal to

$$n_s = n - n_l = n - \frac{k(k-1)}{2} = \frac{2n - k(k-1)}{2}.$$

All largest $n_l = C_k^2$ discs may be transferred from the first peg to the peg B_2 no less than with $4n_l - 2k + 1$ moves (with the help of Theorem 2).

Then for transferring all n discs in the case C) independently of used algorithm $H(Min, k, n)$ moves are needed, where $H(Min, k, n)$ is equal to

$$H(Min, k, n) = 8\frac{2n - k(k-1)}{2} + 4\frac{k(k-1)}{2} - 2k + 1 =$$

$$= 8n - 4k^2 + 4k + 2k^2 - 2k - 2k + 1 = 8n - 2k^2 + 1.$$

For getting the upper estimate of the function $H_k(n)$ we use the next algorithm.

Algorithm A_3

1. Move C_k^2 smallest discs from the first peg to the peg B_k by the algorithm A_2.
2. Move C_{k-1}^2 next (greater according to the size) discs from the first peg to the peg B_{k-1} by the algorithm A_2.
3. Move C_{k-2}^2 next (greater according to the size) discs from the first peg to the peg B_{k-2} by the algorithm A_2.

   At last on stage $k-1$ we move one n-disc from B_1 to B_2.

k. Move all discs from pegs B_3,$B_4, \dots, B_k$ to the peg B_2 by the algorithm A_2.

In this case C) we can transfer the maximum number of discs n, which is equal to

$$n = C_2^2 + C_3^2 + \cdots + C_{k-1}^2 + C_k^2 = C_{k+1}^3.$$

For getting the upper estimate of the function $H_k(n)$ in the case C) we use the following formula

$$H(A_3, k, n) = H_2(n_2) + 2(H_3(n_3) + H_4(n_4) + \cdots + H_k(n_k)),$$

where n_i is a number of discs placed on the peg B_i, where $i \in \{2, \cdots, k\}$ and $n_2 = 1, n_3 = 3$.

We get with the help of formulas (1), (4)

$$H(A_3,k,n) = 1+2(7+(4n_4-2*4+1)+(4n_5-2*5+1)+\cdots+(4n_k-2k+1)) =$$

$$= 1+2(4(n_4+n_5+\cdots+n_k) - \frac{2(k+4)(k-3)}{2} + k - 3 + 7) =$$

$$= 1+2(4(n-4)-k^2+3k-4k+12+k+4) = 1+2(4n-k^2) = 8n-2k^2+1.$$

Then A_3 is the optimal algorithm and in the case C) we have

$$8n-2k^2+1 \le H_k(n) \le 8n-2k^2+1.$$

Theorem 3 is proved.

7. Estimations of the function $H_k(n)$ for the case D).

Theorem 4. If $k \ge 3$ and $C^3_{k+1} < n \le C^4_{k+2}$, then the exact estimation of the function $H_k(n)$ is equal to

$$H_k(n) = 16n - \frac{2k((k+1)(2k+1)-3)}{3} + 1. \tag{6}$$

Proof. Let's infer the under estimate $H(Min,k,n)$ of the function $H_k(n)$ for our case D).

It is impossible (by Theorem 3) to transfer a smallest disc (1-disc) from B_1 to B_2 with 2^3 moves, if $n > C^3_{k+1}$. In the case D) we have n_s discs for transferring of each $2^4 = 16$ moves are needed.

Obviously, the number of such (smallest) discs (in case D)) is equal to

$$n_s = n - C^3_{k+1} = n - \frac{(k+1)k(k-1)}{6}.$$

All largest $n_l = C^3_{k+1}$ discs may be transferred from the first peg to the peg B_2 no less than with $8n_l - 2k^2 + 1$ moves (with the help of Theorem 3).

Then we have

$$H(Min,k,n) = 16(n - \frac{(k+1)k(k-1)}{6}) + 8C^3_{k+1} - 2k^2 + 1 =$$

$$= 16n - 4\frac{(k+1)k(k-1)}{3} - 2k^2 + 1.$$

Then for the case D) we have

$$H(Min,k,n) = 16n - 2(k^2-1)\frac{2k+3}{3} - 1. \tag{7}$$

For getting the upper estimate of the function $H_k(n)$ we use the next algorithm.

Algorithm A_4

1. Move C^3_{k+1} smallest discs from the first peg to the peg B_k by the algorithm A_3. In this order to transfer from the B_1 to the peg B_2 smallest C^2_k discs, to the peg B_3 next C^2_{k-1} discs, ... to the B_k one disc and later move all discs from pegs B_{k-1},B_{k-2}, ... ,B_2 to the peg B_k by the algorithm A_2.
2. Move C^3_k next (greater according to the size) discs from the first peg to the peg B_{k-1} by the algorithm A_3.
3. Move C^3_{k-1} next (greater according to the size) discs from the first peg to the peg B_{k-2} by the algorithm A_3.

At last on stage $k-1$ we move one n-disc from B_1 to B_2.

k. Move all discs from pegs B_3,B_4, ... , B_k to the peg B_2 by the algorithm A_3.

In this case D) we can transfer the maximum number of discs n , where

$$n = C^3_3 + C^3_4 + \cdots + C^3_k + C^3_{k+1} = C^4_{k+2}.$$

For getting the upper estimate of the function $H_k(n)$ in the case D) we use the following formula

$$H(A_4, k, n) = H_3(n_3) + 2(H_4(n_4) + H_5(n_5) + \cdots + H_k(n_k)),$$

where n_i means a number of discs placed on the peg B_i, where $i \in \{3, \cdots , k\}$.

With the help of formulas (1), (5) and with the condition $n_3 = 5$ we obtain

$$H(A_4, k, n) = 31+2((8n_4-2*4^2+1)+(8n_5-2*5^2+1)+\cdots+(8n_k-2*k^2+1)) =$$

$$= 31 + 16(n-5) - 4\sum_{i=4}^{k} i^2 + 2(k-3) =$$

$$= 16n + 31 - 80 + 56 + 2(k-3) - \frac{2k(k+1)(2k+1)}{3} =$$

$$= 16n + 7 + \frac{6(k-3) - 2k((k+1)(2k+1))}{3} =$$

$$= 16n + 7 + \frac{2k((k+1)(2k+1) - 6(k-3)}{3} =$$

$$= 16n + 7 - 6 - \frac{2k((k+1)(2k+1) - 3)}{3} =$$

$$= 16n + 1 - \frac{2k((k+1)(2k+1) - 3)}{3}.$$

So we get

$$H(A_4,k,n) = 16n + 1 - \frac{2k((k+1)(2k+1)-3)}{3}. \qquad (8)$$

Now we will compare our under estimate (7) with our upper estimate (8). We have

$$H(A_4,k,n) - H(Min,k,n) = 16n + 1 - \frac{2k((k+1)(2k+1)-3)}{3} - 16n +$$

$$+2(k^2-1)\frac{2k+3}{3} + 1 = 2(k^2-1)\frac{2k+3}{3} - \frac{2k((k+1)(2k+1)-3)}{3} + 2 =$$

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

$$= \frac{2(2k^3+3k^2-2k-3) - 2k(2k^2+3k-2)+6}{3} =$$

$$= \frac{2k(2k^2+3k-2) - 6 - 2k(2k^2+3k-2)+6}{3} = 0.$$

Therefore, for the case D) $H(Min,k,n) = H(A_4,k,n)$ and our algorithm A_4 is the optimal algorithm. Then the exact estimate for the case D) is equal to

$$H_k(n) = 16n - \frac{2k((k+1)(2k+1)-3)}{3} + 1 = 16n - 2(k^2-1)\frac{2k+3}{3} - 1.$$

Theorem 4 is proved.

8. Conclusions

Our method for getting of under estimates of the function $H_k(n)$ may be generalized for any cases, where $k > 3$ and

$$C_{k+t-3}^{t-1} < n \le C_{k+t-2}^{t}.$$

The parameter t may be defined always with the help of the Pascal's triangle for concrete values n and k.

We use from [5] the formula

$$tri_d(n+1) = \sum_{i=1}^{n} tri_{d-1}(i),$$

where $tri_d(j)$ is a "d-triangle" number.

It is known $tri_1(j) = C_j^1 = j$ natural numbers, $tri_2(j) = C_j^2 = 1 + 2 + \cdots + (j-1)$ triangular numbers, $tri_3(j) = C_j^3$ tetrahedral (pyramidal) numbers, which are the sum of consecutive triangular numbers.

We can write this formula as

$$C_{n+1}^d = \sum_{i=1}^{n} C_i^{d-1}.$$

Then our under estimate for the function $H_k(n)$ is equal to

$$H(Min, k, n) = 2^t(n - C_{k+t-3}^{t-1}) + \sum_{i=0}^{t-1} 2^i C_{(k-3)+i}^i. \quad (9)$$

Formula (9) allows to obtain values $H(Min, k, n)$ for any n and k. In particular $H(Min, 4, 64) = 18433, H(Min, 5, 64) = 1535, H(Min, 6, 64) = 673$, which coincide with corresponding values from [3]. Howevere this fact does not prove that our formula (9) is the exact estimate of the function $H_k(n)$.

References

[1] N. Claus (= É. Lucas). La Tour d'Hanoi, Jeu de Calcul, Science et Nature, 1, (8), 127—128, 1884.

[2] P.K. Stockmeyer. The Tower of Hanoi: A Bibliography, September, 2005.

[3] S. Novikov. Upper estimates of complexity of algorithms for multi-peg Tower of Hanoi problem, Annales Academiae Paedagogicae Cracoviensis, Studia Mathematica VI, 57-66, 2007.

[4] S. Klavzar, U. Milutinovic, C. Petr. On the Frame-Stewart algorithm for the multi-peg Tower of Hanoi problem, Discrete Applied Mathematics, 120, 141—157, 2002.

[5] http:// en.wikipedia.arg/Pascal 27s triangle