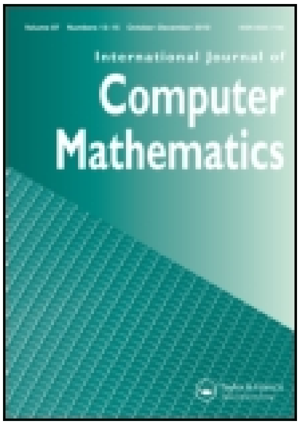

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

A recursive algorithm for the multi-peg tower of hanoi problem

M. Kaykobad [a] , S.T. Rahman [a] , R-.A. Bakhtiar [b] & A.A.K. Majumdar [c]
[a] Department of Computer Science and Engineering, BUET, Dhaka, 1000, Bangladesh
[b] Onirbaan Group, Dhaka, Bangladesh
[c] Department of Mathematics, Jahangirnagar University, Savor, Dhaka, 1342, Bangladesh
Version of record first published: 19 Mar 2007.

To cite this article: M. Kaykobad, S.T. Rahman, R-.A. Bakhtiar & A.A.K. Majumdar (1995): A recursive algorithm for the multi-peg tower of hanoi problem, International Journal of Computer Mathematics, 57:1-2, 67-73

To link to this article: http://dx.doi.org/10.1080/00207169508804412

Intern. J. Computer Math., Vol. 57, pp. 67–73
Reprints available directly from the publisher
Photocopying permitted by license only

A RECURSIVE ALGORITHM FOR THE MULTI-PEG TOWER OF HANOI PROBLEM

M. KAYKOBAD, S. T-.U. RAHMAN

Department of Computer Science and Engineering, BUET, Dhaka-1000, Bangladesh

R-.A. BAKHTIAR

Onirbaan Group, Dhanmondi, Dhaka, Bangladesh

A. A. K. MAJUMDAR

Department of Mathematics, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh

(Received 22 June 1994)

This paper gives a recursive algorithm to solve the multi-peg Tower of Hanoi problem. The algorithm is based on the dynamic programming equation satisfied by the optimal value function, $M(n, p)$, where $M(n, p)$ denotes the minimum number of moves required to solve the problem with n discs and p pegs. This algorithm is the only one available, particularly for the case when $p \geqslant 5$.

KEY WORDS: Algorithms, Multi-Peg Tower of Hanoi, minimum partition numbers, recurrence relations.

C.R. CATEGORIES: F.2.2, G.2.1.

AMS (1991) SUBJECT CLASSIFICATION: Primary 90C31.

1 INTRODUCTION

The multi-peg Tower of Hanoi problem, posed by Lucas in 1889, is as follows: $p(\geqslant 3)$ pegs are fastened to a stand, and $n(\geqslant 1)$ discs of different radii, $D_1, D_2, \ldots, D_n$, initially rest on the source peg, P_1, in a tower in small-on-large ordering. The object is to transfer the tower to the destination peg, P_p, in minimum number of moves, under the condition that each move can shift only the topmost disc from any peg to another such that no disc is ever placed on top of a smaller one. The 3-peg problem is a popular puzzle of mathematical recreation. It has also gained some popularity as an example of recursive procedure in many computing text books, such as Reingold and Hansen [7] and Aho, Hopcroft and Ullman [1].

Let us call the problem with n discs and p pegs the (n, p)-system, and let $M(n, p)$ be the minimum number of moves required to solve it. Then, the dynamic programming equation satisfied by $M(n, p)$ is given (see, for example, Wood [8] and Hinz [4]) by

$$M(n, 3) = 2M(n-1, 3) + 1, \quad n \geqslant 2, \tag{1.1}$$

$$M(n, p) = \min_{1 \leqslant k \leqslant n-1} \{2M(k, p) + M(n-k, p-1)\}, \quad n \geqslant 2, \quad p \geqslant 4, \tag{1.2}$$

$$M(1, p) = 1 \quad \text{for all} \quad p \geqslant 3 \tag{1.3}$$

Let $l(n, p)$ minimize the expression on the right-hand side of (1.2). The number $l(n, p)$ would be called a minimum partition number for the given (n, p)-system. Thus, in order to solve a given (n, p)-system with $p \geqslant 4$ optimally, we have to find an optimal number $l(n, p)$ of discs which should be transferred to an intermediate peg using all the p pegs, then move optimally the remaining $(n - l(n, p))$ discs on P_1 to the destination peg, P_p, using the $(p-1)$ pegs available, and finally, move the $l(n, p)$ discs on the intermediate peg to P_p, optimally, again using the p pegs. Thus, under the optimal strategy described above, the problem of solving the given (n, p)-system reduces to the problem of finding a number $l(n, p)$, and then solving associated subsystems, $(l(n, p), p)$ and $(n - l(n, p), p-1)$.

This paper solves the (n, p)-system by finding upper and lower bounds of $l(n, p)$. Since no explicit expression for $l(n, p)$ is available in literature till now, our algorithm is the only one to deal with the multi-peg problem (with $p \geqslant 5$). We have also derived an expression for $M(n, p)$, which matches with that given by Frame [3], but his analysis is confined only to finding $M(n, p)$.

2 OPTIMAL STRATEGY

Let $M_j(n, p)$ be the number of moves required to transfer the jth disc, D_j, $(j = 1, 2, \ldots, n)$ from P_1 to P_p under the optimal strategy.

LEMMA 2.1 *For all* $j = 1, \ldots, n$, *and* $p \geqslant 3$,

$$M_j(n, p) = 2^{k_j(n,p)}$$

for some integer $k_j(n, p) \geqslant 0$.

Proof The result holds true for $p = 3$ with

$$k_j(n, 3) = n - j \quad \text{for} \quad n \geqslant 1,\ j = 1, 2, \ldots, n.$$

Also, obviously

$$k_j(1, p) = 0 \quad \text{for all} \quad p \geqslant 3.$$

To prove the lemma by double induction on n and p, let us assume its validity for some n and p.

Now, according to the argument leading to the equation (1.2), the $(n+1,\ p)$-system is subdivided into the two subsystems $(l(n+1,\ p),\ p)$ and $(n+1-l(n+1,\ p),\ p-1)$, and D_j belongs to one of these subsystems. In either case, the result follows by the induction hypothesis.

Similarly, the $(n,\ p+1)$-system is subdivided into the subsystems $(l(n,\ p+1),\ p+1)$ and $(n-l(n,\ p+1),\ p)$. If D_j belongs to the second subsystem, the result follows by the induction hypothesis. On the other hand, if D_j belongs to the first subsystem, then we have

$$M_j(n,\ p+1) = 2M_j(l(n,\ p+1),\ p+1)$$

If the right-hand side is unfolded further, we will find D_j either in the second subsystem, in which case the result would hold true by the induction hypothesis, or else, D_j will belong to the first subsystem with $(p+1)$ pegs and a smaller number of discs. It may be noted that, each time D_j falls into the first subsystem, we have a multiplication factor of 2. Continuing this, eventually, either we will come up with D_j in the second subsystem, or we will have the single disc D_j in the first subsystem which requires $2^o = 1$ move to reach p_p.

This completes the induction. □

For any $r \in N = \{0, 1, 2, \ldots\}$, let $N(r,\ p)$ be the maximum number of discs each of which reaches P_p in exactly 2^r moves using all the p pegs under the optimal strategy, and let $S(n,\ p)$ be the maximum number of discs each of which can be shifted to P_p in no more than 2^r moves using the p pegs under the optimal strategy. Note that, in defining $N(r,\ p)$ and $S(r,\ p)$, we do not restrict ourselves to particular values of n.

LEMMA 2.2 *For all* $r \geqslant 1$ *and* $p \geqslant 4$,

$$N(r,\ p) = N(r,\ p-1) + N(r-1,\ p)$$

with

$$N(0,\ p) = 1 \quad \text{for all} \quad p \geqslant 3,$$

$$N(r,\ 3) = 1 \quad \text{for all} \quad r \geqslant 0.$$

Proof The boundary conditions are quite obvious. To determine the recurrence relation satisfied by $N(r,\ p)$, we observe that the discs each of which requires 2^r moves to reach P_p are

(1) those which reach P_p in 2^r moves without visiting a particular intermediate peg (and hence, the total number of pegs available in this case is $p-1$),

(2) those which reach some intermediate peg in 2^{r-1} moves and then reach P_p.

These give respectively the first and second terms on the right-hand side of the recurrence relation. □

The following lemma would prove fruitful.

LEMMA 2.3 (1) *For all* $l, m \in \mathbb{N}$

$$\sum_{j=0}^{l} \binom{j+m}{m} = \binom{l+m+1}{m+1}$$

(2) *For all* $r \in \mathbb{N}$ *and* $p \geqslant 3$,

(*a*)
$$N(r, p) = \binom{p+r-3}{p-3},$$

(*b*)
$$S(r, p) = \binom{p+r-2}{p-2},$$

(*c*)
$$N(r, p) = S(r, p-1),$$

(*d*)
$$S(r+1, p+1) = S(r+1, p) + S(r, p+1)$$

Now, given an (n, p)-system, let $r(n, p)$ be the minimum value of r satisfying the inequality

$$M_j(n, p) \leqslant 2^{r(n,p)} \quad \text{for all} \quad j = 1, 2, \ldots, n; \tag{2.1}$$

also, let $N_m(r(n, p), p)$ be the maximum number of discs each of which requires exactly $2^{r(n,p)}$ moves to reach P_p from P_1 under the optimal strategy, so that

$$N_m(r(n, p), p) = n - S(r(n, p) - 1, p). \tag{2.2}$$

Then we have the following lemma which is evident from (2.1).

LEMMA 2.4 *For all* $n \geqslant 2$ *and* $p \geqslant 3$,

$$S(r(n, p) - 1, p) < n \leqslant S(r(n, p), p).$$

The right-hand side inequality of Lemma 2.4 states that, n cannot be greater than the maximum number of discs each of which can be moved to P_p in no more than $2^{r(n,p)}$ moves using the p pegs, and the left-hand side inequality expresses the fact that n must be greater than the maximum number of discs each of which requires no more than $2^{r(n,p)-1}$ moves to reach P_p using p pegs.

It may be mentioned here that

$$r(n, 3) = n - 1 \quad \text{for all} \quad n \geqslant 1. \tag{2.3}$$

The following lemma gives bounds on $l(n, p)$.

LEMMA 2.5 For all $n \geqslant 2$ *and* $p \geqslant 4$,

(1) $n - S(r(n, p), p-1) \leqslant l(n, p) \leqslant S(r(n, p) - 1, p)$,
strict equality signs would hold if and only if $n = S(r(n, p), p)$,

(2) $S(r(n, p)-2, p) \leqslant l(n, p) \leqslant n-S(r(n, p)-1, p-1)$.

Proof (1) Since $l(n, p)$ is the number of discs stored on some intermediate peg (under the optimal strategy), each such disc reaches P_p in no more than $2^{r(n,p)-1}$ moves using all the p pegs. This gives the right-hand side inequality.

Similarly, the $n-l\,(n, p)$ discs still lying on P_1, would each require no more than $2^{r(n,p)}$ moves to reach P_p, and the available pegs are $p-1$ in number, so that

$$n-l\,(n, p) \leqslant S(r(n, p), p-1).$$

This prove the left-hand side inequality.

To prove the other part, let

$$n-S\,(r(n, p), p-1)=l(n, p)=S(r(n, p)-1, p).$$

Then, by Lemma 2.3 (2d),

$$n-S(r(n, p)-1, p)+S(r(n, p)p-1)=S(r(n, p), p).$$

Conversely, if $n=S(r(n, p)-1, p)$, then since

$$n-S(r(n, p), p-1)=S(r(n, p)-1, p),$$

we get

$$S(r(n, p)-1, p-1) \leqslant l(n, p) \leqslant S(r(n, p)-1, p)$$
$$\Rightarrow l(n, p)=S(r(n, p)-1, p)$$

(2) First note that

$$n-l(n, p) \leqslant N_m(r(n, p), p)+S(r(n, p)-1, p-1), \tag{2.4}$$

since some of the $n-l(n, p)$ discs lying on P_1 would be shifted to P_p in $2^{r(n,p)}$ moves to reach P_p using the available pegs.

Similarly,

$$l(n, p) \leqslant N_m(r(n, p), p)+S(r(n, p)-2, p). \tag{2.5}$$

Now, from (2.4), using (2.1) and Lemma 2.3 (2d), we get

$$l(n, p) \geqslant S(r(n, p)-1, p)-S(r(n, p)-1, p-1)=S(r(n, p)-2, p),$$

which is the left-hand side inequality. Again, from (2.5), together with (2.2) and Lemma 2.3 (2d), we have

$$l(n, p) \leqslant n-[S(r(n, p)-1, p)-S(r(n, p)-2, p)]=n-S(r(n, p)-1, p-1),$$

giving the right-hand inequality.

Thus the lemma is proved. □

Combining parts (1) and (2) of Lemma 2.5, we have the following main result of the paper.

THEOREM 2.1 *For $n \geqslant 2$ and $p \geqslant 4$, $k(n, p) \leqslant l(n, p) \leqslant K(n, p)$, where*

$$k(n, p) = \max(S(r(n, p) - 2, p), n - S(r(n, p), p - 1)),$$
$$K(n, p) = \min(n - S(r(n, p) - 1, p - 1), S(r(n, p) - 1, p)).$$

The above theorem is now exploited to devise a recursive algorithm to any (n, p)-system with $p \geqslant 4$. The algorithm is as follows:

Algorithm Multi-Peg Tower of Hanoi (n, p, S, D)

```
/* n-disks, p-pegs, S-source peg, D-destination peg */
/* l(n, p)-optimal no. of disks to be placed on intermediate peg*/
  if (n = 1) then
    shift top disk from S to D
  else
    Multi_Peg_Tower_of_Hanoi (l(n, p), p, S, P_i)
/* P_i is an intermediate peg on which the smallest of the l(n, p) disks can be shifted
without violating the ordering rule */
    Multi_Peg_Tower_of_Hanoi (n - l(n, p), p - 1, S, D)
    Multi_Peg_Tower_of_Hanoi (l(n, p), p, P_i, D)
  endif
end algorithm
Algorithm Multi_peg_Tower_of_Hanoi
```

It may be mentioned here that the algorithms of Lu[6], Hinz[4], Chu and Johnsonbaugh [2], and Liefvoort [5] are for the 4-peg problem only.

We now have the following theorem giving an expression for $M(n, p)$.

THEOREM 2.2 *For $n > 2$ and $p > 3$,*

$$M(n, p) = 2^{r(n,p)}\left(n - \binom{p + r(n, p) - 3}{p - 2}\right) + \sum_{t=0}^{r(n,p)-1} 2^t \binom{p + t - 3}{p - 3}$$

where,

$$S(r(n, p) - 1, p) < n \leqslant S(r(n, p), p).$$

Proof From (2.2), we get

$$n = N_m(r(n, p), p) + \sum_{t=0}^{r(n,p)-1} N(t, p).$$

Now, each of the $N_m(r(n, p), p)$ discs requires exactly $2^{r(n,p)}$ moves to reach P_p from P_1, under optimal strategy, and each of the $N(t, p)$ discs would require 2^t moves ($t = 0, 1, \ldots, r(n, p) - 1$) to reach P_p from P_1. Therefore,

$$M(n, p) = 2^{r(n,p)} N_m(r(n, p), p) + \sum_{t=0}^{r(n,p)-1} 2^t N(t, p).$$

Now, appealing to (2.2) and Lemma 2.3(2a), the theorem follows. □

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *Data Structures and Algorithms* (Addison-Wesley Book Co., Reading, MA, 1983).
[2] I-Ping Chu and R. Johnsonbaugh, The four-peg Tower of Hanoi Puzzle, *SIGCSE Bulletin*, **23**(3) (1991), 2–4.
[3] J. S. Frame, Solution to AMM problem 3918, *Amer. Math. Monthly*, **48** (1941), 216–219.
[4] A. M. Hinz, An iterative algorithm for the Tower of Hanoi with four pegs, *Computing*, **42** (1989), 135–140.
[5] A. Van de Liefvoort, An iterative algorithm for the Reve's puzzle, *The Computer J.*, **35** (1) (1992), 91–92.
[6] Xue-Miao Lu, An iterative solution for the 4-peg Towers of Hanoi, *The Computer J.*, **35** (2) (1989), 187–189.
[7] E. M. Reingold and W. J. Hansen, *Data Structures*, (Little Brown & Co., NY, 1983).
[8] D. Wood, Towers of Brahma and Hanoi revisited, *J. Recreational Math.*, **14** (1) (1981–82) 17–24.