The many towers of Hanoi

Alan Dix
alan@hcibook.com

http://www.hcibook.com/alan/papers/hanoi93/

The problem

Many years ago in the foothills of the Himalayas, there was a lonely monastery, and in it were three crystal towers on which were sixty-four golden rings of different sizes. Year by year the monks move the rings between the towers following the well known rules, and as is also well known when the task is finished the monks will sit back contented and the world disappear in a puff of smoke. Excitement mounted in the monastery, they had been at the task for four thousand years, and now over half of the disks had been moved. Recruitment was up, pilgrims and novices flocked to the monastery and across the valley their rival monastery looked on in despair. Then as the story goes, a wandering mathematician appeared and after scribbling on the back of a yak for a few moments declared that the world would be safe for another 450 billion years. Within days the pilgrims were dispersed and the towers fell into disrepute.

The abbot of the other monastery was not unduely alarmed at the demise of his rival, but saw that his own monastery was in need of a little publicity. If they should by chance find some towers of their own... But how many towers should there be? Too many towers would be a trifle expensive (have you ever tried to make a crystal tower), and besides if the task were to be finished in a few weeks and the world not end, what would people think? Yet if there were too few, and some inconvenient mathematician calculated it would take millions of years to complete, it would lack popular appeal. What was needed was a task taking the odd thousand years, short enough to arouse interest, and long enough to hide away before it became a liability.

Happily the mathematician was still around, and the abbot made discrete enquiries. The backs of all the yaks in the monastery were covered in calculations and even a few passing yeti had been recruited to the task when the mathematician reported back to the abbot...

The algorithm

A class of algorithms is posited which is thought to be optimal, but I have no proof of this, however out of this class, the optimal algorithm has been found, and thus an upper bound for the number of moves necessary has been found.

The class of algorithms posited for solving the many towers of Hanoi is as follows:

1. If there are only three towers, use the traditional algorithm
2. Choose some number t less than n, the number of rings, but possibly zero.
3. Move the top n-t rings to the last tower, using this algorithm.
4. Move the remaining t rings onto the target tower using the remaining towers and this algorithm.
5. Move the smallest n-t rings back from the last tower to the target tower using this algorithm again.

There are of course recursively many choices of t's throughout the algorithm, and one will try to make an optimal choice. The optimal algorithm of this class can then be seen to yield h_n^m moves where m+2 is the number of towers, and:

$$h_n^m = \min_{t \in \{1..n\}} 2h_{n-t}^m + h_t^{m-1}$$

Experiment

We can solve this recurrence relation directly for small values of m and n and we get, starting with the familiar powers of two solution for three towers:

0,1,3,7,15,31,63,...
0,1,3,5,9,13,17,25,33,41,...
0,1,3,5,7,11,15,19,23,27,31,...
0,1,3,5,7,9,13,17,21,25,29,...
0,1,3,5,7,9,11,15,19,23,27,...

The beginning of the series clearly settles down, and this corresponds to the choice of t as n-1, that is using all the towers when there are very few rings, after that the pattern is less obvious, however if we look at difference, we get:

1,2,4,8,16,32,64,...
1,2,2,4,4,4,8,8,8,8,...
1,2,2,2,4,4,4,4,4,4,8,..
1,2,2,2,2,4,4,4,4,4,4,...
1,2,2,2,2,2,4,4,4,4,4,...

Extending these series shows that the differences are always the powers of two in sequence, but with larger repeats as one moves down and to the right. The number of repeats is as follows:

1,1,1,1,1,1,1,1,...
1,2,3,4,5,...
1,3,6,10,...
1,4,10,..
1,5,..

An hypothesis

The r^{th} term in each series is the sum of the first r terms in the series above, that is these series satisfy:

$$S_r^m = \sum_{k=1}^{r} S_k^{m-1}$$

$$S_k^1 = 1$$

$$S_r^m = \frac{r(r+1)(r+2)...(r+m-1)}{m!}$$

The "key" points in the original series, where the difference changes its power of two would be given by:

$$SP_r^m = \sum_{k=1}^{r} S_k^m 2^{k-1}$$

$$SP_r^m = 2^r (S_r^m - S_r^{m-1} + S_r^{m-2} - ...)$$

And the actual series is H_n^m where:

$$H_n^m = SP_r^m + i\, 2^r$$

where $n = S_r^m + i\$$

and $S_r^m <= n <= S_{r+1}^m$

Several formulae will be useful about these series :

$$H^m_{n+1} - H^m_n = 2^r$$

$$S^m_r <= n <= S^m_{r+1}$$

Which follows immediately from the above definition and:

LEMMA

$$SP^m_r = 2\, SP^m_{r-1} + SP^{m-1}_r$$

PROOF

$$\begin{aligned} RHS &= 2 \sum_{k=1}^{r} S^m_{k-1}\, 2^{k-2} + \sum_{k=1}^{r} S^{m-1}_k\, 2^{k-1} \\ &= \sum_{k=1}^{r} (S^m_{k-1} + S^{m-1}_k)\, 2^{k-1} \\ &= \sum_{k=1}^{r} S^m_k\, 2^{k-1} = LHS \end{aligned}$$

Proving the hypothesis

The above series certainly matches the solutions for h^m_n for small values, but does it extend to larger values? We need to prove the hypothesis:

THEOREM

$$h^m_n = H^m_n$$

PROOF

We proceed by induction of m and n, we assume that it is true for:

(a) all m, n' such that n' < n

(b) all m', n such that m' < m

We first prove the base cases:

Base cases

two cases (i) m any value and n=0, and (ii) m=1

first case m any value and n=0, no towers, so:

$$h^m_0 = 0 = H^m_0$$

QED (i) n=0

second case m=1

$h^1_n = 2^n - 1$ standard algorithm

$S^1_n = n \quad \Rightarrow \quad H^1_n = SP^1_n = 2^n - 1$

QED (ii) m=1

Inductive case

$$h^m_n = \min_{t \in \{1..n\}} 2H^m_{n-t} + H^{m-1}_t$$

$$= \min_{t \in \{1..n\}} f(t)$$

where $f(t) = 2H^m_{n-t} + H^{m-1}_t$

We look at the differences f(t+1)-f(t):

$$f(t+1)-f(t) = (2H^m_{n-t-1} + H^{m-1}_{t+1}) - (2H^m_{n-t} + H^{m-1}_t)$$

$$= (H^{m-1}_{t+1} - H^{m-1}_t) - 2(H^m_{n-t} - H^m_{n-t-1})$$

$$= 2^r - 2\times 2^s$$

where $S^{m-1}_r <= t <= S^{m-1}_{r+1}$

and $S^m_s <= n\text{-}t\text{-}1 <= S^m_{s+1}$

r increases with t and s decreases with t, f(t) is therefore convex.

There are two possible behaviours for f:
- (i) f has a flat patch where r =s-1
- (ii) f has a unique minimum point

We deal with each case in turn.

Case (i) f has a flat patch where r =s-1

$\exists$ t, r **st.** r = s-1 **and**

$S^{m-1}_r <= t <= S^{m-1}_{r+1}$

$t = S^{m-1}_r + i$

$S^m_{r-1} <= n\text{-}t <= S^m_r$

$n\text{-}t = S^m_{r-1} + j$

$\Rightarrow$ $S^m_r < n < S^m_{r+1}$

$n = S^m_r + i + j$

so $h^m_n = 2H^m_{n-t} + H^{m-1}_t$

$$= 2(SP^m_{r-1} + j\,2^{r-1}) + (SP^{m-1}_r + i\,2^r)$$

$$= 2\,SP^{m}_{r-1} + SP^{m-1}_{r} + (i+j)\,2^{r}\,)$$

$$= 2\,SP^{m}_{r} + (i+j)\,2^{r}\,)$$

$$= H^{m}_{n}$$

QED case (i)

Case (ii) f has a unique minimum point

$\exists$ t **st.** f(t+1) - f(t) +ve **and** f(t) - f(t-1) -ve

i.e. $r(t) > s(t)+1$ **and** $r(t-1) < s(t-1)+1$

$\Rightarrow$ $r(t) > s(t)+2$

(as r and s only change by one)

also

$$t = S^{m-1}_{r}$$

$$n-t-1 = S^{m}_{s+1} - 1$$

$$n-t = S^{m}_{s+1} = S^{m}_{r-1}$$

$$\Rightarrow \quad n = S^{m}_{r}$$

so

$$h^{m}_{n} = 2\,H^{m}_{n-t} + H^{m-1}_{t}$$

$$= 2\,SP^{m}_{r-1}\ \ SP^{m-1}_{r}$$

$$= 2\,SP^{m}_{r}$$

$$= H^{m}_{n}$$

QED case (ii)

QED inductive cases

QED theorem

Approximate values

From the formulae for S^{m}_{r} and SP^{m}_{r}, we see that

$$r \approx (m!\,n)^{1/m}$$

$$\approx m \times n^{1/m}$$

$$\log_2 h^{m}_{n} \approx (1 - 1/m)\log_2 n + m \times n^{1/m}$$

The later being very much an order of magnitude approximation.

To do

I have obtained a formula for, and proved the correctness of the optimal

many towers of Hanoi. The class of algorithms seem sensible, but I have not proved
that it is optimal as against other possible algorithms. Two ways forward are suggested:

- Exhaustively obtain the optimal solution for small numbers of towers and rings, and check against the formulae obtained.

- Analytically prove that the class of algorithms (and therefore the optimal within that class) is optimal over all choices of moves.

I am persuing the former course initially and if the results are favourable will then proceed to the latter option. It does however seem a hard problem on both counts because of the sheer size of the search space.