CHAPTER 1

Recurrent Problem

RECURRENT PROBLEM

Reference Book:

Concrete Mathematics – Graham, Knuth, Patashnik

Topics:

The Tower of Hanoi

Lines in the Plane

The Josephus Problem

LEARNING OUTCOME

Define Recurrent Problems

Determine Recurrence Equations

Learn Difference between Upper and Lower Bounds

Learn Different Proof Techniques

Find Closed form of the Recurrence Equations

TOWER OF HANOI

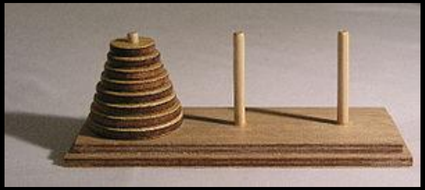
The Tower of Hanoi (also called the Tower of Brahma or Lucas' Tower and sometimes pluralized) is a mathematical game or puzzle.

It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

I. Only one disk can be moved at a time.
II. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
III. No disk may be placed on top of a smaller disk.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is 2^n − 1, where n is the number of disks.



LET'S TRY SOME SMALL CASE FIRST

DETERMINE RECURRENCE EQN OF TOH

Let's say that T_n is the minimum number of moves that will transfer n disks from one peg to another under Lucas's rules. Then T_1 is obviously 1,

and $T_2 = 3$.

Remember, $T_0 = 0$

What if bigger case arrives!!!

DETERMINE RECURRENCE EQN OF TOH

We first transfer the n − 1 smallest to a different peg (requiring Tn−1 moves), then move the largest (requiring one move), and finally transfer the n−1 smallest back onto the largest (requiring another Tn−1 moves).

Thus we can transfer n disks (for $n > 0$) in at most $2T_{n-1} + 1$ moves:

$$T_n \leqslant 2T_{n-1} + 1, \qquad \text{for } n > 0.$$

This formula uses ` $\leq$ ' instead of ` $=$ ' because our construction proves only that $2T_{n-1} + 1$ moves **suffice**; we haven't shown that $2T_{n-1} + 1$ moves are necessary.

DETERMINE RECURRENCE EQN OF TOH

At some point, we must move the largest disk.

When we do, the n − 1 smallest must be on a single peg, and it has taken at least Tn−1 moves to put them there.

We might move the largest disk more than once, if we're not too alert.

But after moving the largest disk for the last time, we must transfer the n−1 smallest disks (which must again be on a single peg) back onto the largest; this too requires Tn−1 moves. Hence----

$$T_n \geqslant 2T_{n-1} + 1, \qquad \text{for } n > 0.$$

DETERMINE RECURRENCE EQN OF TOH

Recurrence Equation from two inequalities:

❖These two inequalities, together with the trivial solution for n = 0, yield

$$T_0 = 0;$$
$$T_n = 2T_{n-1} + 1, \qquad \text{for } n > 0.$$

CLOSED FORM

The recurrence allows us to compute T_n for any n we like.

But nobody really likes to compute from a recurrence, when n is large; it takes too long.

The recurrence only gives indirect, local information.

A solution to the recurrence would make us much happier.

That is, we'd like a nice, neat, “closed form" for T_n that lets us compute it quickly, even for large n.

CLOSED FORM

One way is to **guess** the correct solution, then to prove that our guess is correct. And our best hope for guessing the solution is to look (again) at small cases.

Proof Technique: Now, we shall use **Mathematical Induction**

So we compute, successively,

$T_3 = 2\cdot3 + 1 = 7$; $T_4 = 2\cdot7 + 1 = 15$; $T_5 = 2\cdot15 + 1 = 31$;

$T_6 = 2\cdot31 + 1 = 63$.

GUESS: It certainly looks as if

$$\boxed{T_n = 2^n - 1, \qquad \text{for } n \geq 0.}$$

MATHEMATICAL INDUCTION

Mathematical induction is a general way to prove that some statement about the integer n is true for all $n>n_0$.

❖First we prove the statement when n has its smallest value, n_0; this is called the basis.

❖Then we prove the statement for $n > n_0$, assuming that it has already been proved for all values between n_0 and $n-1$, inclusive; this is called the induction.

❖Such a proof gives infinitely many results with only a finite amount of work.

CLOSED FORM PROOF

Basis:

For n=0, $T_0 = 2^0 - 1 = 0$.

So, for basis, its true.

Induction: (for n>0)

Let, the closed form is true for n-1

i.e., $T_{n-1} = 2^{n-1} - 1$

Now, we have to prove the closed form for n=n

Hence, $T_n = 2\,T_{n-1} + 1$

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

$= 2^{n-1+1} - 2 + 1$

$= 2^n - 1$

According to mathematical induction the closed form is proved for all $n \geq 0$

QUESTION

A double Tower of Hanoi contains 2n disks of n different sizes, two of each size. As usual, we are required to move only one disk at a time, without putting a larger disk over a smaller one. How many moves does it take to transfer a double tower from one to another, if the ordering of the disks of equal size is not important? Follow the following steps:

(a) Describe your algorithm and from there, five an upper bound.
(b) Prove that your upper bound is also a lower bound.
(c) Then find the closed form of the recursive equation by any method you like.