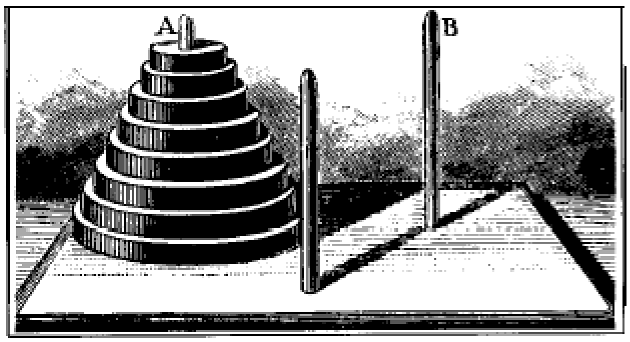
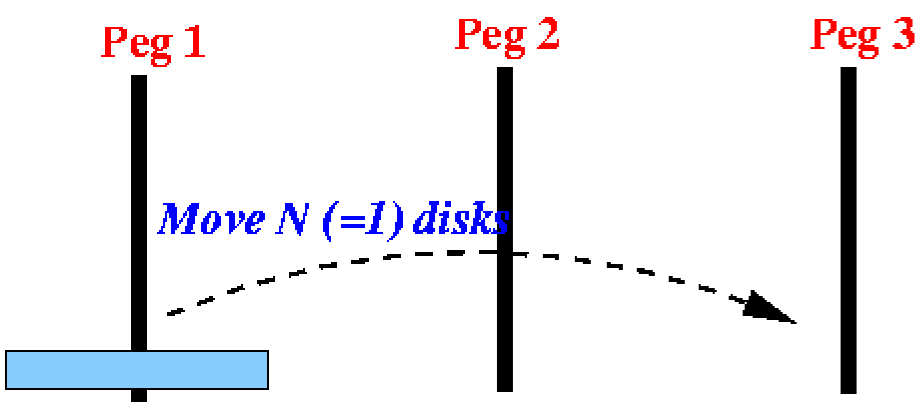
Tower of Hanoi:

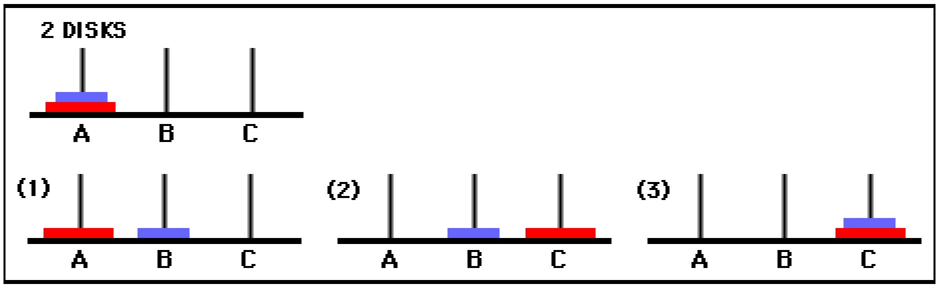
Let's look first at a neat little puzzle called the Tower of Hanoi, invented by the French mathematician Edouard Lucas in 1883. We are given a tower of eight disks, initially stacked in decreasing size on one of three pegs. The objective is to transfer the entire tower to one of the other pegs (here, from A to B), moving only one disk at a time and never moving a larger one onto a smaller.

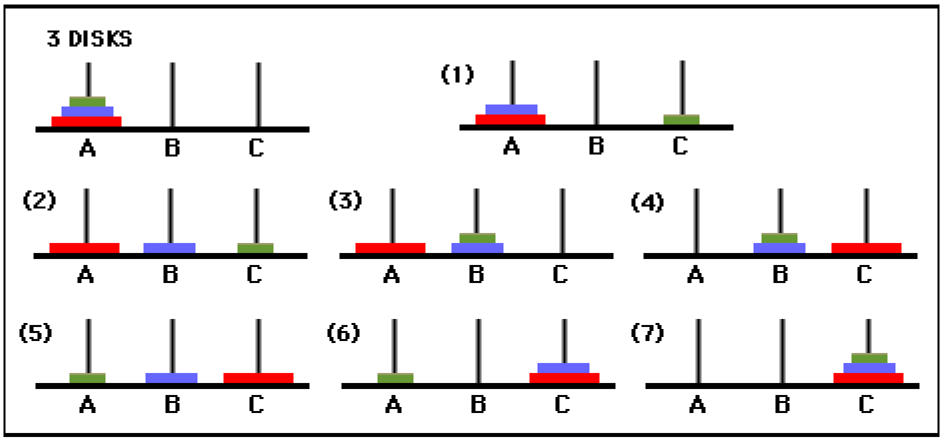


It's not immediately obvious that the puzzle has a solution. Now the question arises: What's the best we can do? That is, how many moves are necessary and sufficient to perform the task? The best way to tackle a question like this is to generalize it a bit. The Tower of Brahma has 64 disks and the Tower of Hanoi has 8; let's consider what happens if there are n disks.

One advantage of this generalization is that we can scale the problem down even more. In fact, we'll see repeatedly in this book that it's advantageous to LOOK AT SMALL CASES first. It's easy to see how to transfer a tower that contains only one or two disks. And a small amount of experimentation shows how to transfer a tower of three.







Let’s say that $T(n)$ is the minimum number of moves that will transfer n disks from one peg to another. Clearly $T(0) = 0$, because no moves at all are needed to transfer a tower of $n = 0$ disks. Then $T(1)$ is obviously 1, and $T(2) = 3$.

Experiments with three disks show that the winning idea is to transfer the top two disks to the middle peg, then move the third, then bring the other two onto it. This gives us a clue for transferring n disks in general: We first transfer the $n - 1$ smallest to a different peg (requiring $T(n-l)$ moves), then move the largest (requiring one move), and finally transfer the $n-1$ smallest back onto the largest (requiring another $T(n-1)$ moves). Thus we can transfer n disks (for $n > 0$) in at most $2T(n-1)+1$ moves.

$T(n) >= 2T(n-1)+1$ for $n>0$

So, the recurrence relation for the tower of Hanoi problem would be defined as follows –

$T(0)=0$

$T(n)= 2T(n-1)+1$ for $n>0$

A set of equalities like this is called a recurrence (a.k.a. recurrence relation or recursion relation). It gives a boundary value and an equation for the general value in terms of earlier ones. 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. So, we will now represent a “closed form” of $T(n)$ which will that let us compute it quickly, even for large n. This “closed form” is basically a mathematical expression or formula. This mathematical expression is known as solution of recursive relation. Here, we will find the solution using iterative method.

$T(n)=2T(n-1)+1$

$=2[2T(n-2)+1]+1$

$=2^2T(n-2)+2+1$

$=2^2[2T(n-3)+1]+2+1$

$=2^3T(n-3)+2^2+2+1$

.

.

$=1+2+2^2+2^3T(n-3)$

$=2^0+2^1+2^2+2^3T(n-3)$

$=2^0+2^1+2^2+\ldots..+2^{n-1}T[n-(n-1)]$

$=2^0+2^1+2^2+\ldots..+2^{n-1}T(1)$

$=2^0+2^1+2^2+\ldots..+2^{n-1}x1$ [as moving one disk requires minimum 1 move, T(1)=1]

$=2^0+2^1+2^2+\ldots..+2^{n-1}$

$= 1(1-2^n)/1-2$ [geometric series, $a (1-r^n)/(1-r)$]

$= 2^n-1$

The solution for the recursive relation of tower of hanoi is 2^n-1