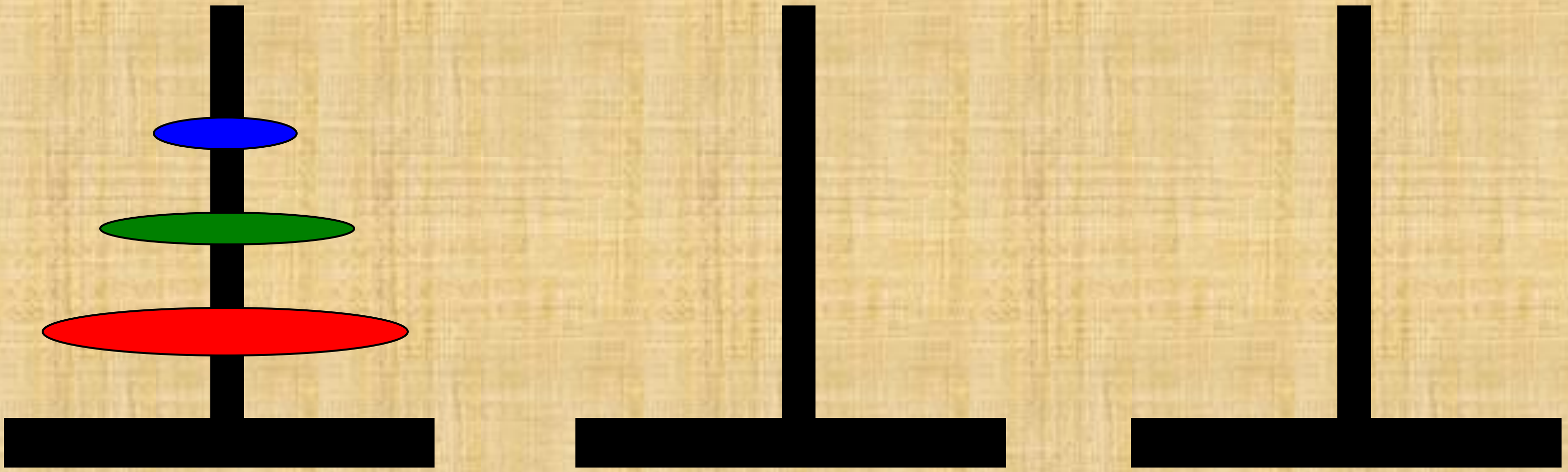
Tower of Hanoi

- There are three towers
- 64 gold disks, with decreasing sizes, placed on the first tower
- You need to move all of the disks from the first tower to the last tower
- Larger disks can not be placed on top of smaller disks
- The third tower can be used to temporarily hold disks

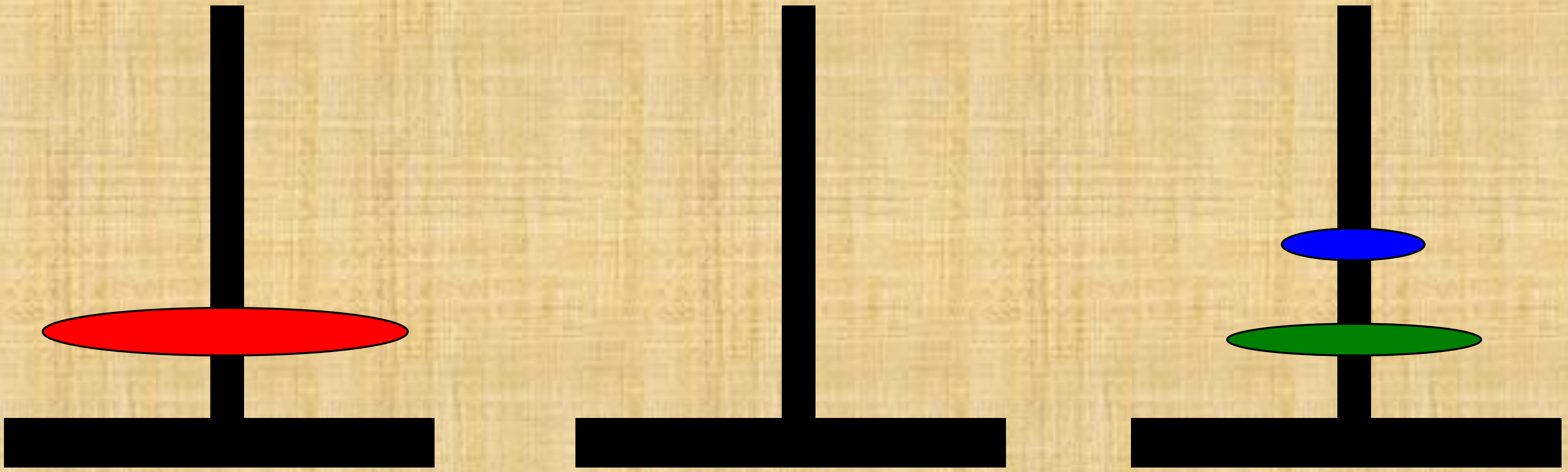
Tower of Hanoi

- The disks must be moved within one week. Assume one disk can be moved in 1 second. Is this possible?
- To create an algorithm to solve this problem, it is convenient to generalize the problem to the “N-disk” problem, where in our case N = 64.

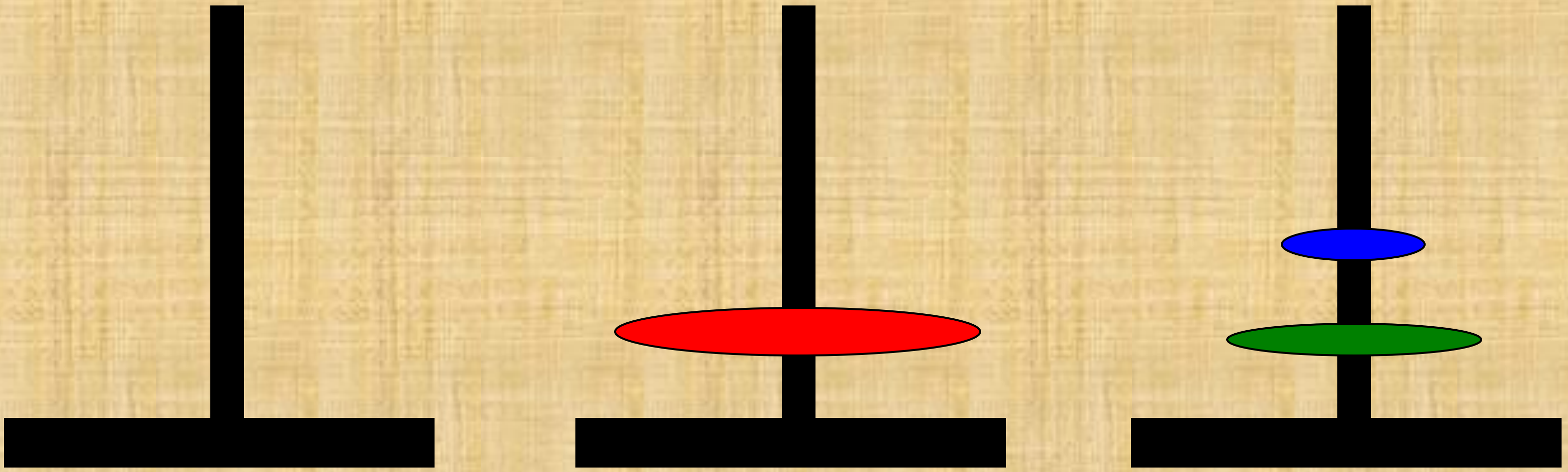
Recursive Solution



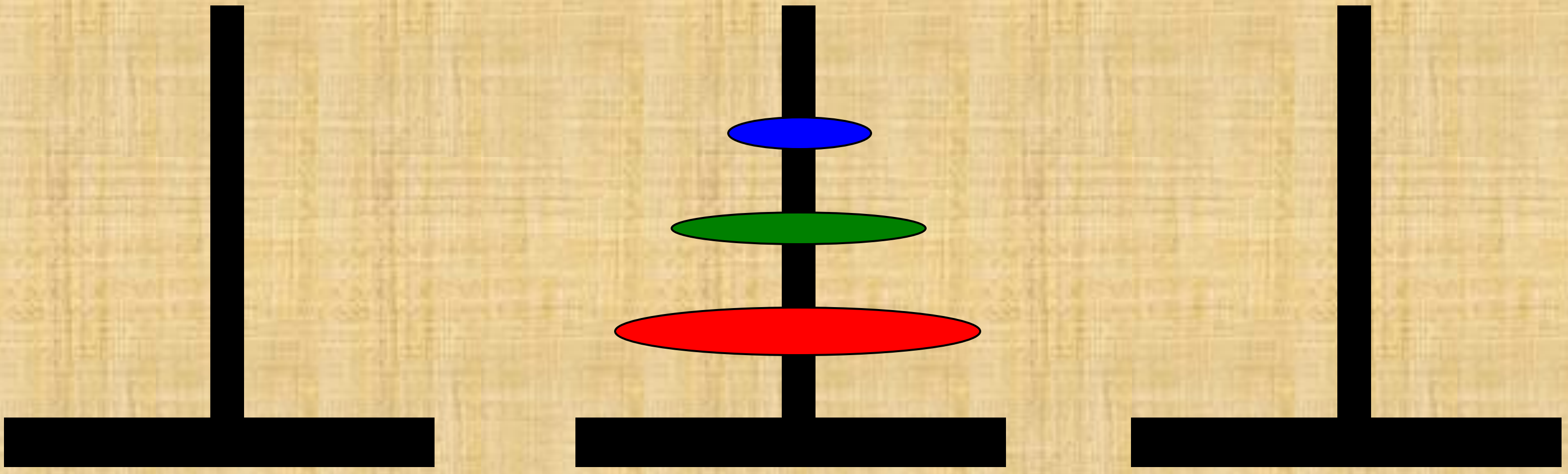
Recursive Solution



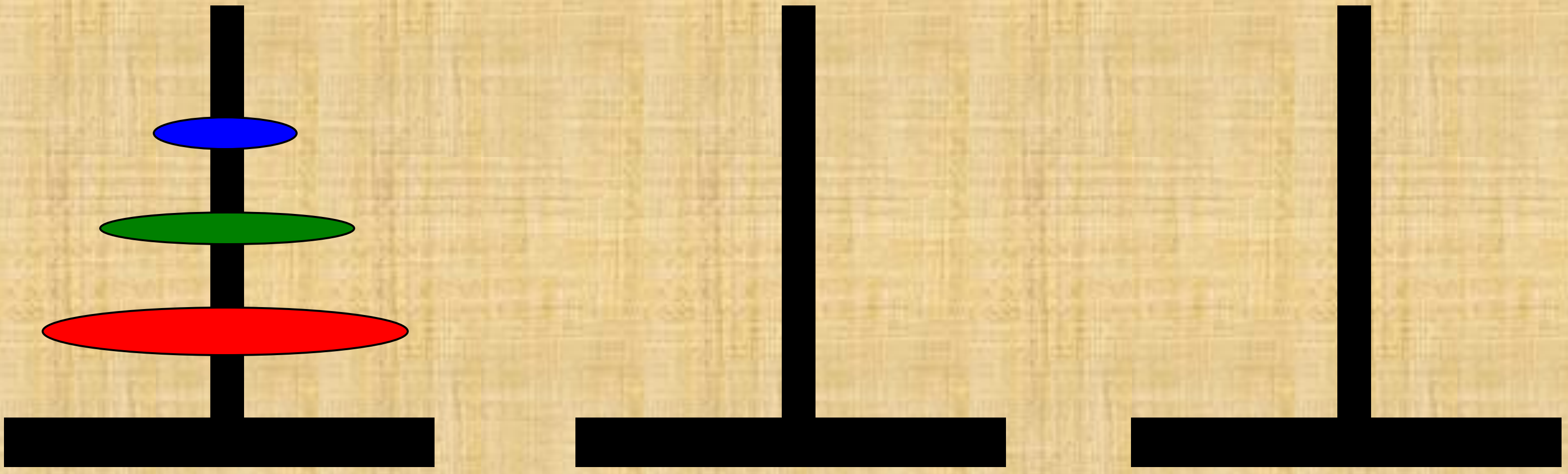
Recursive Solution



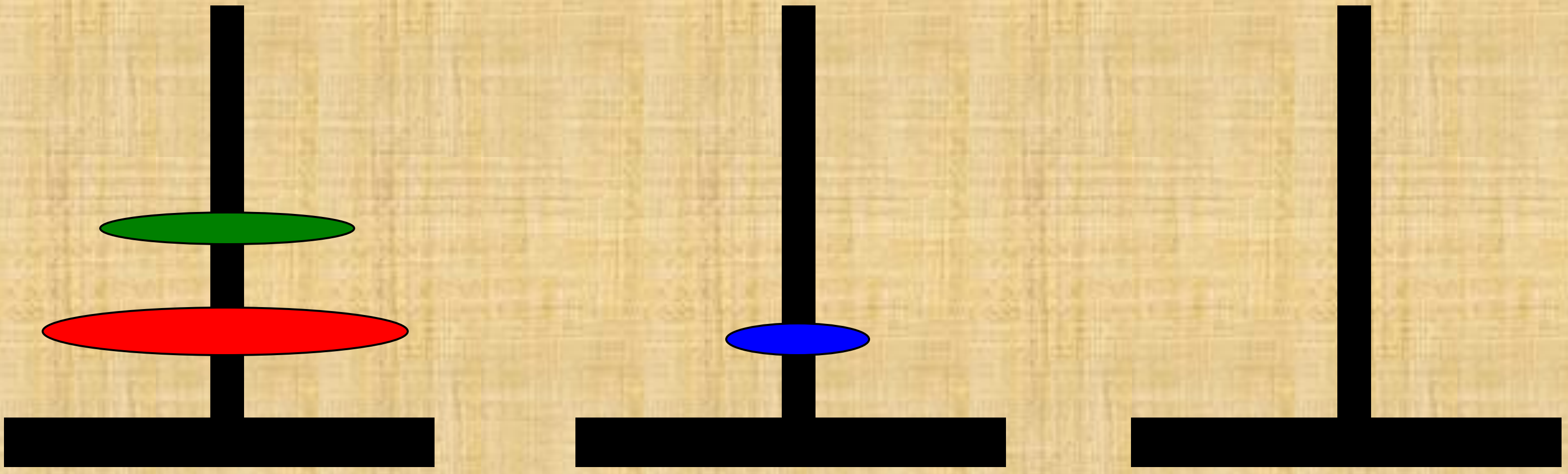
Recursive Solution



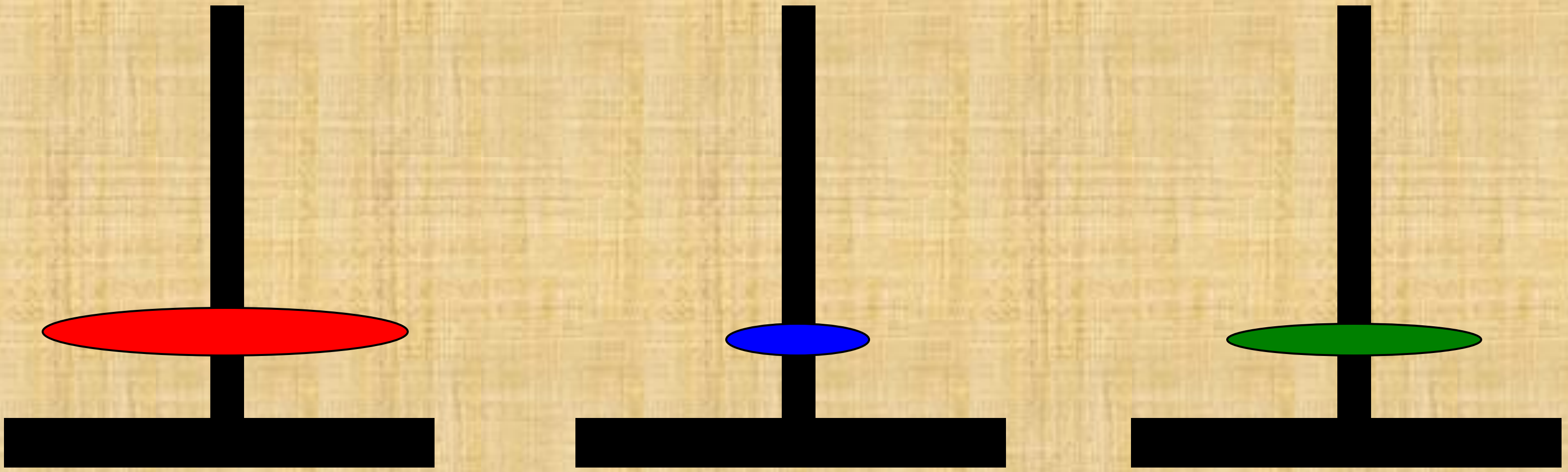
Tower of Hanoi



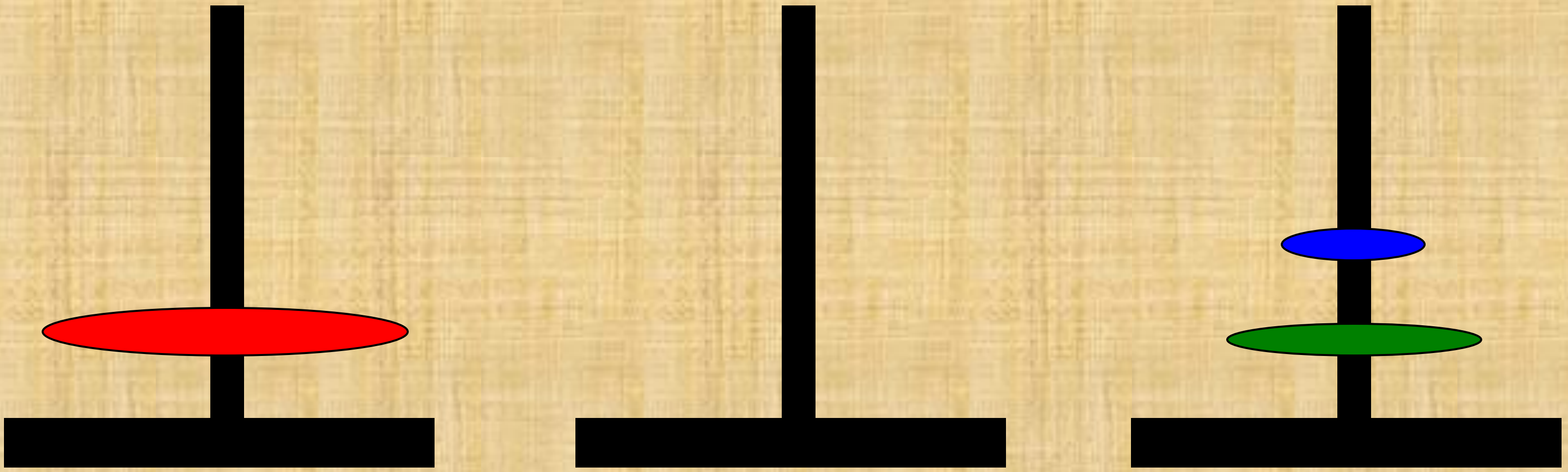
Tower of Hanoi



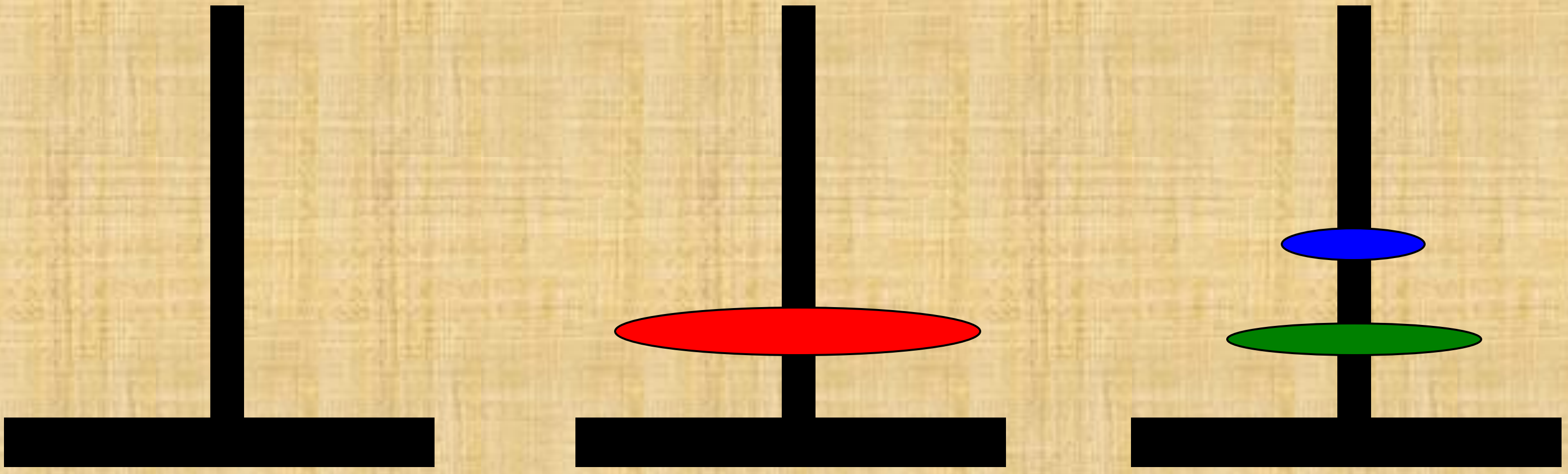
Tower of Hanoi



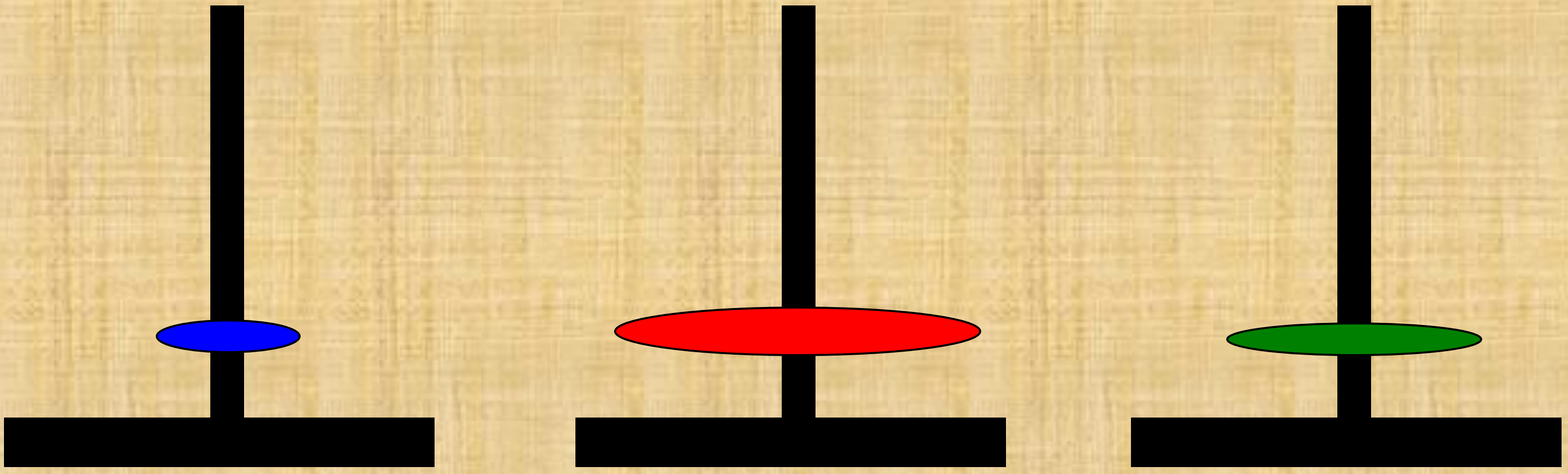
Tower of Hanoi



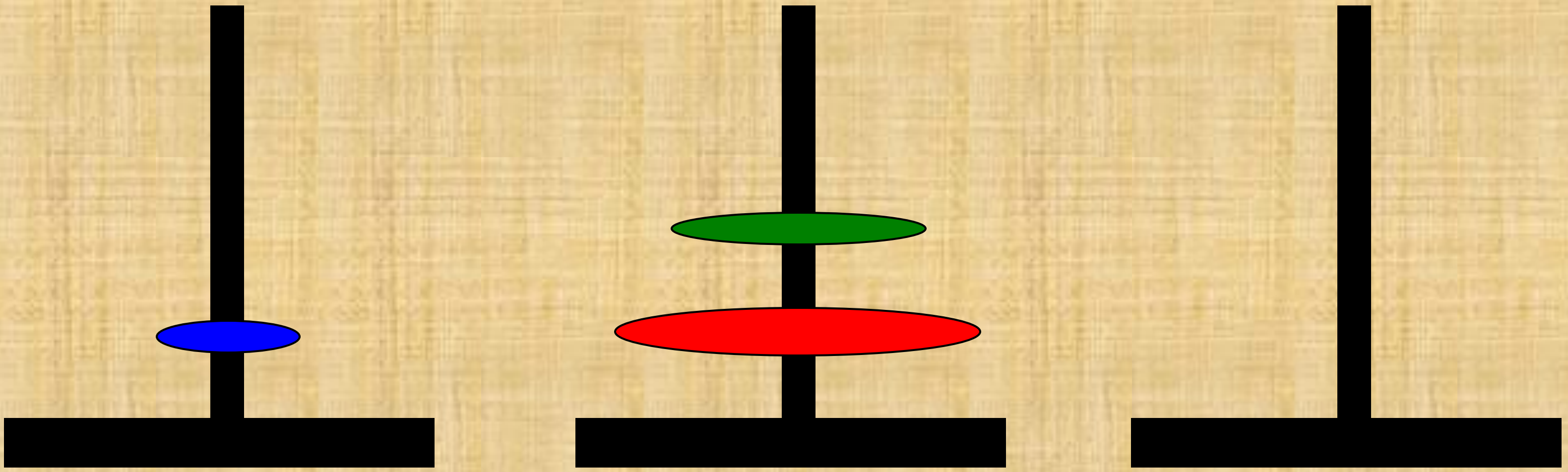
Tower of Hanoi



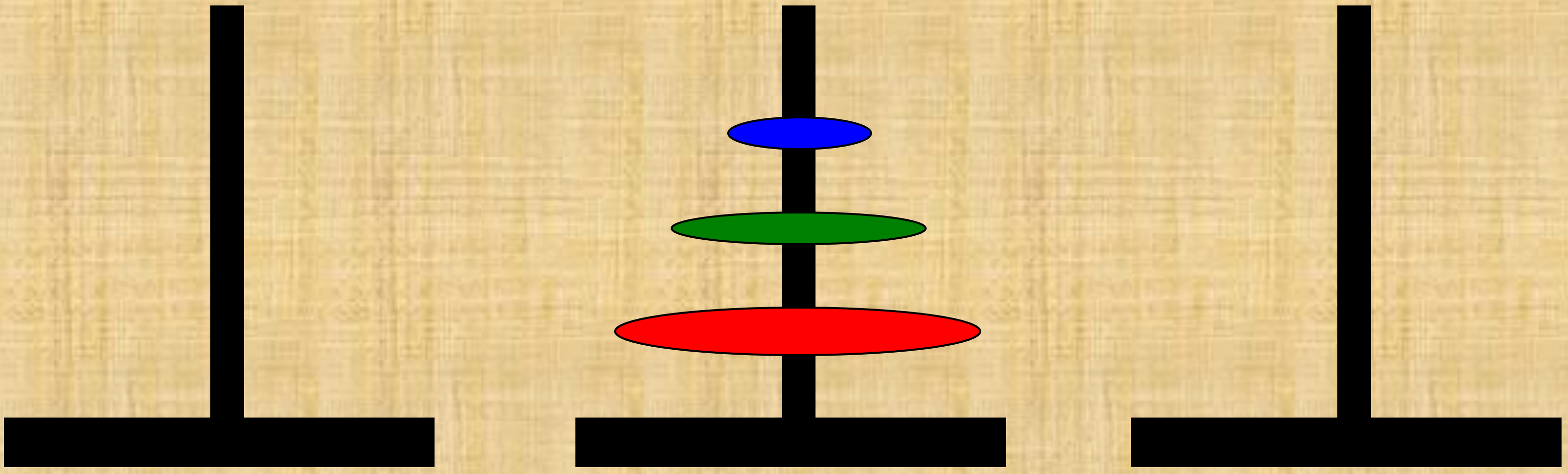
Tower of Hanoi



Tower of Hanoi



Tower of Hanoi



Recursive Algorithm

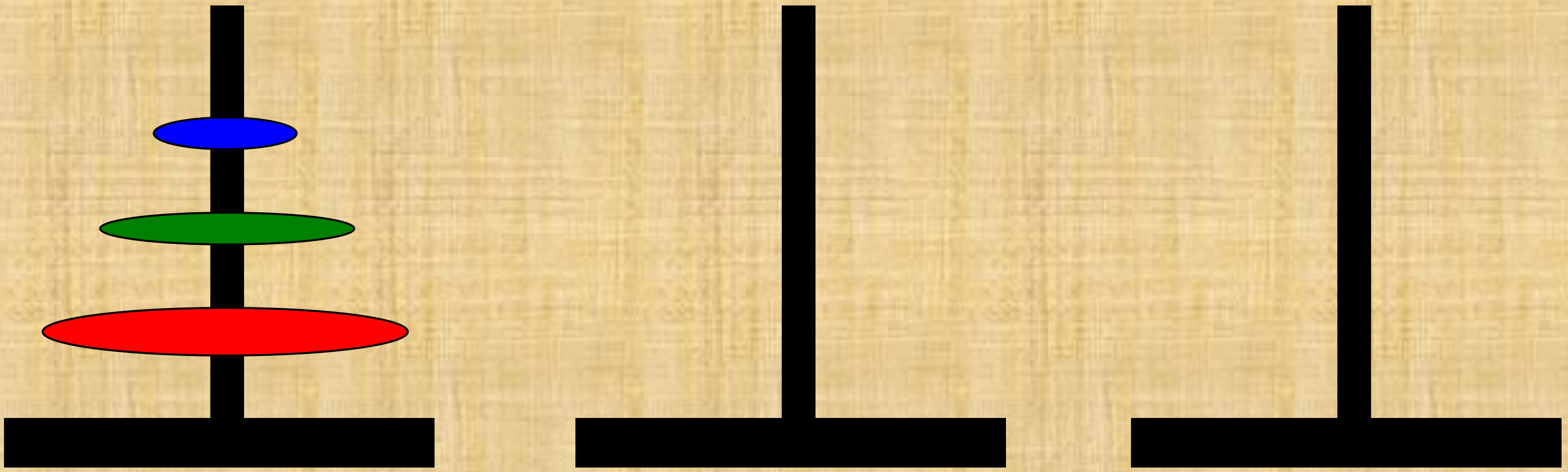
```
void Hanoi(int n, string a, string b, string c)
  {
    if (n == 1)  /* base case */
      Move(a,b);
    else { /* recursion */
      Hanoi(n-1,a,c,b);
      Move(a,b);
      Hanoi(n-1,c,b,a);
    }
  }
```

Iterative Formula

- Let $G_n(i)$ be a function from $[0,\ldots,2^n-1]$
- $G_n(i) = i \wedge (i >> 1)$ [exclusive or of i and i/2]
  - $G_2(0) = 0$, $G_2(1) = 1$, $G_2(2) = 3$, $G_2(3) = 2$
- Use induction to prove that the sequence $G_n(i)$, $i=0,\ldots,2^n-1$ is a binary-reflected Gray code.

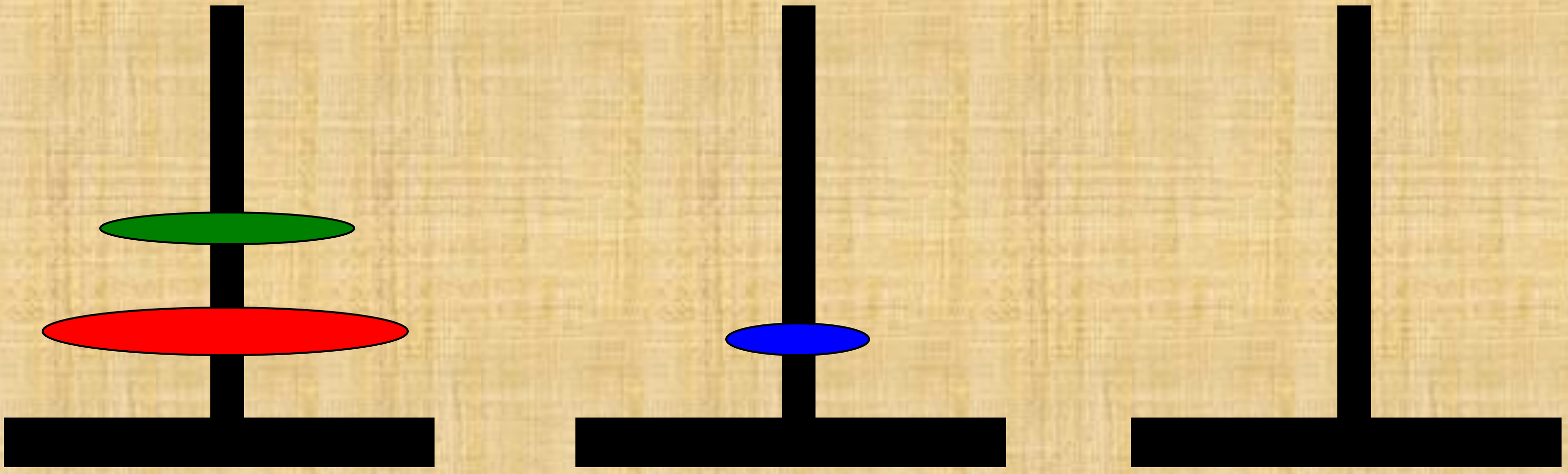
Tower of Hanoi

(0,0,0)



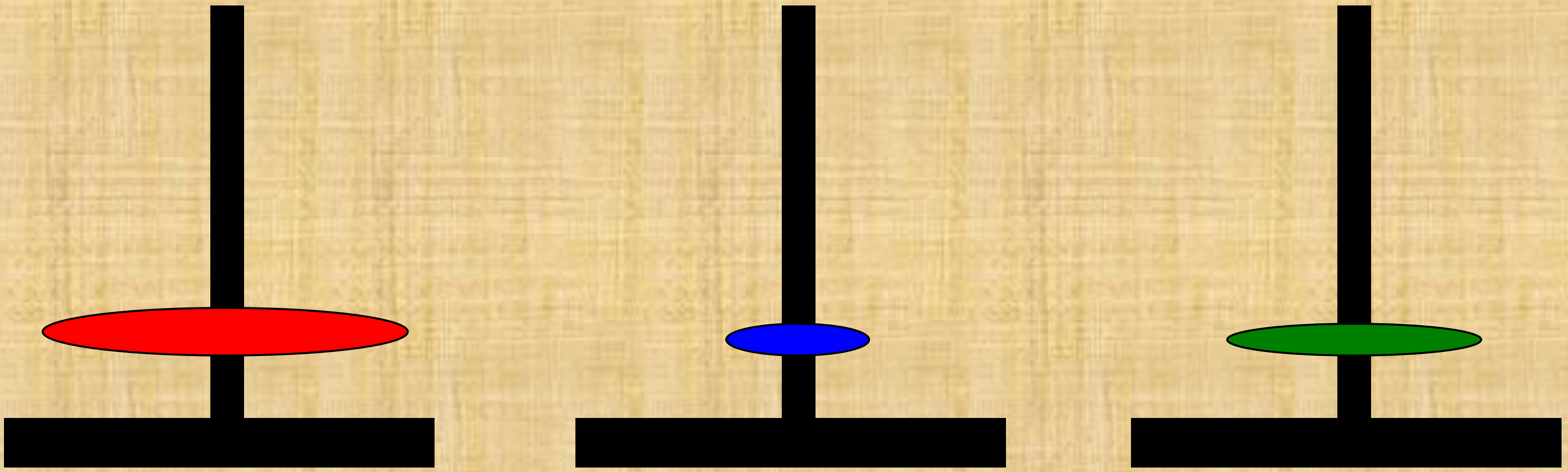
Tower of Hanoi

(0,0,1)



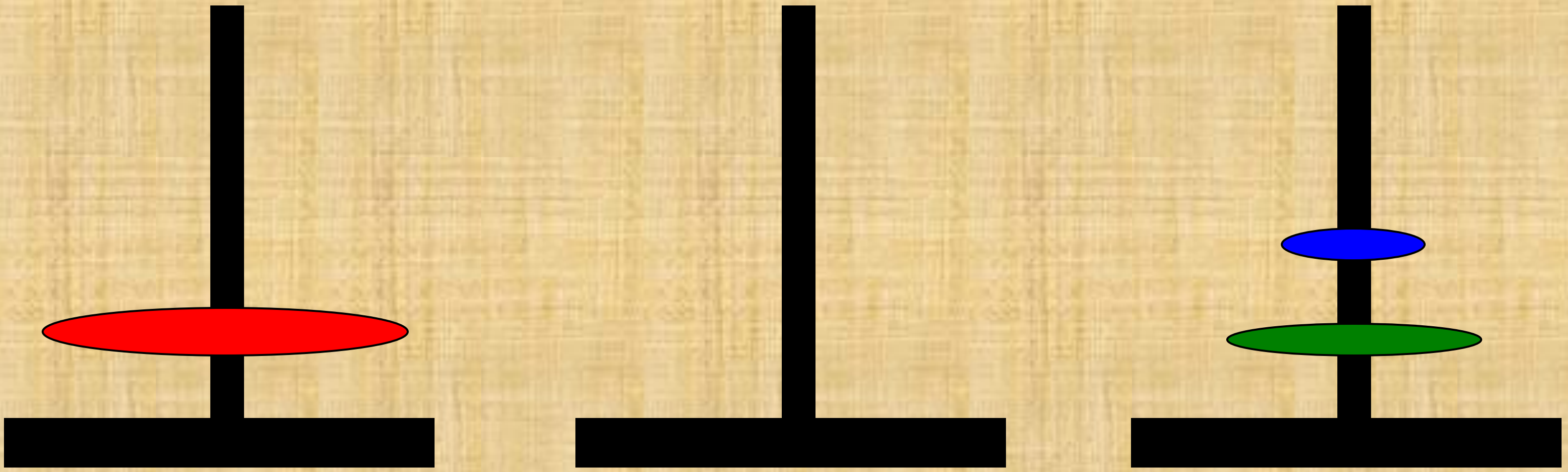
Tower of Hanoi

(0,1,1)



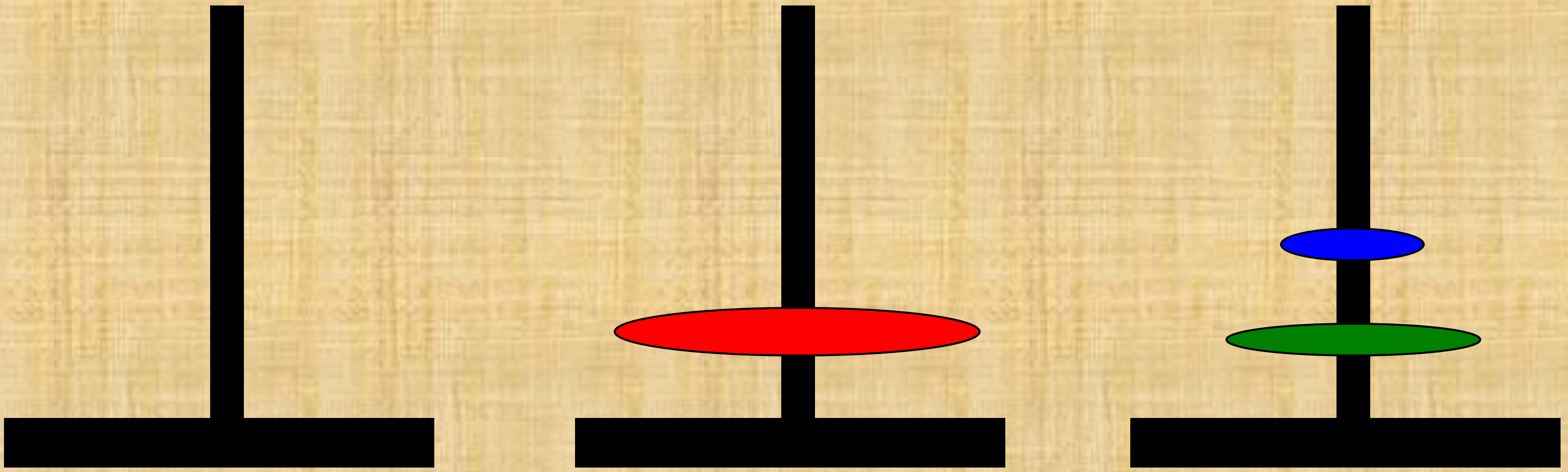
Tower of Hanoi

(0,1,0)



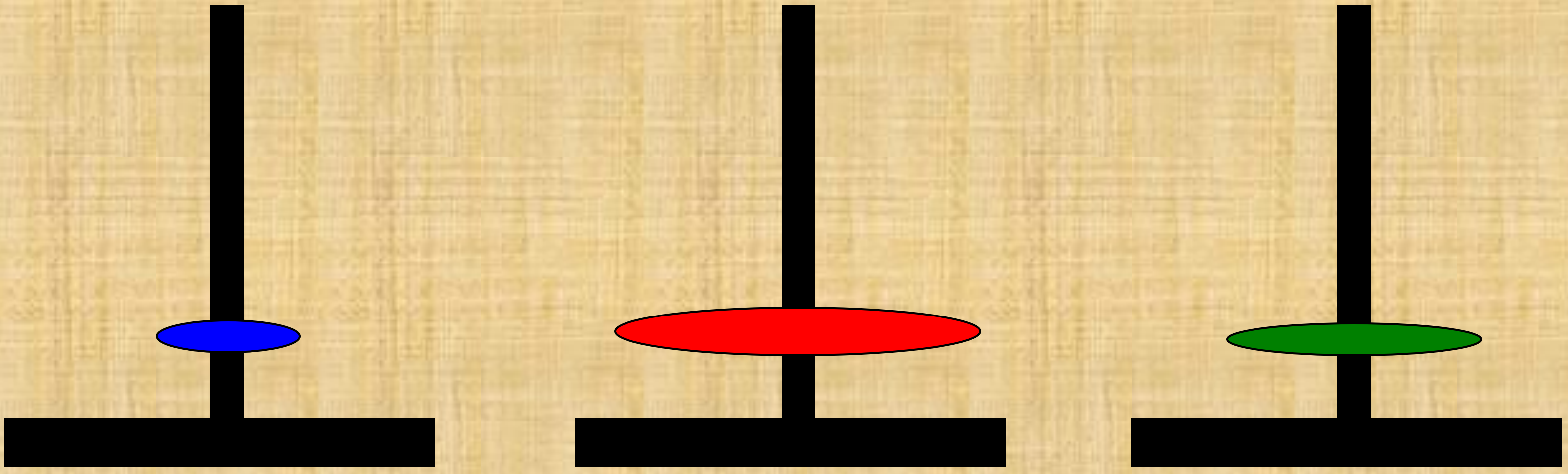
Tower of Hanoi

(1,1,0)



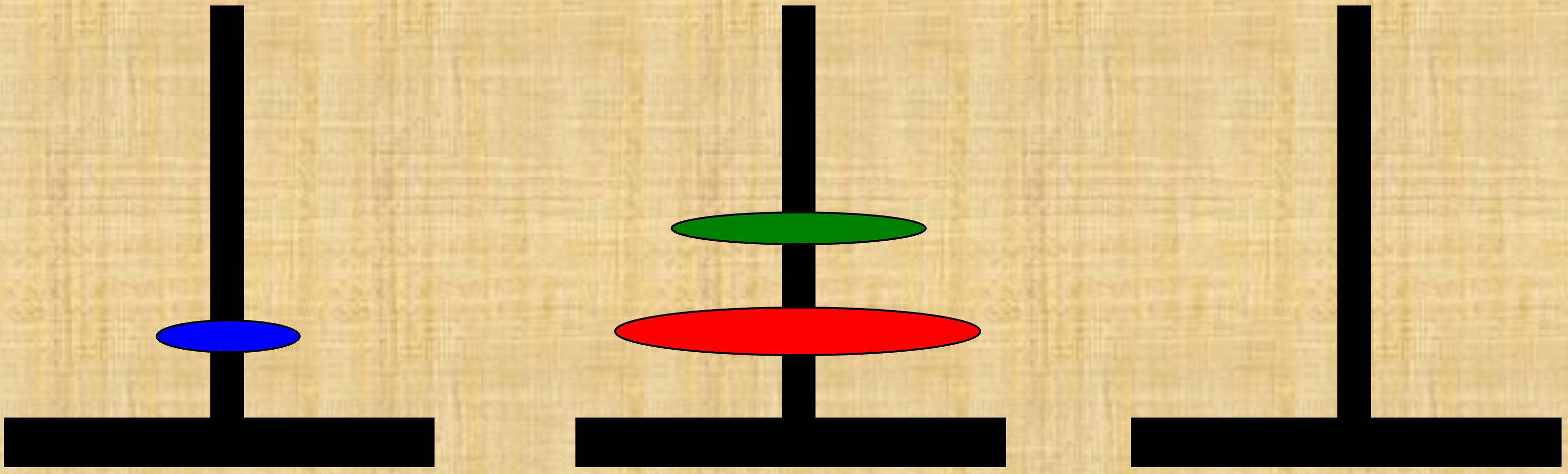
Tower of Hanoi

(1,1,1)



Tower of Hanoi

(1,0,1)



Tower of Hanoi

(1,0,0)