Recursion and Induction

- Themes
  - Recursion
  - Recurrence Relations
  - Recursive Definitions
  - Induction (prove properties of recursive programs and objects defined recursively)
- Examples
  - Tower of Hanoi
  - Gray Codes
  - Hypercube

Tower of Hanoi

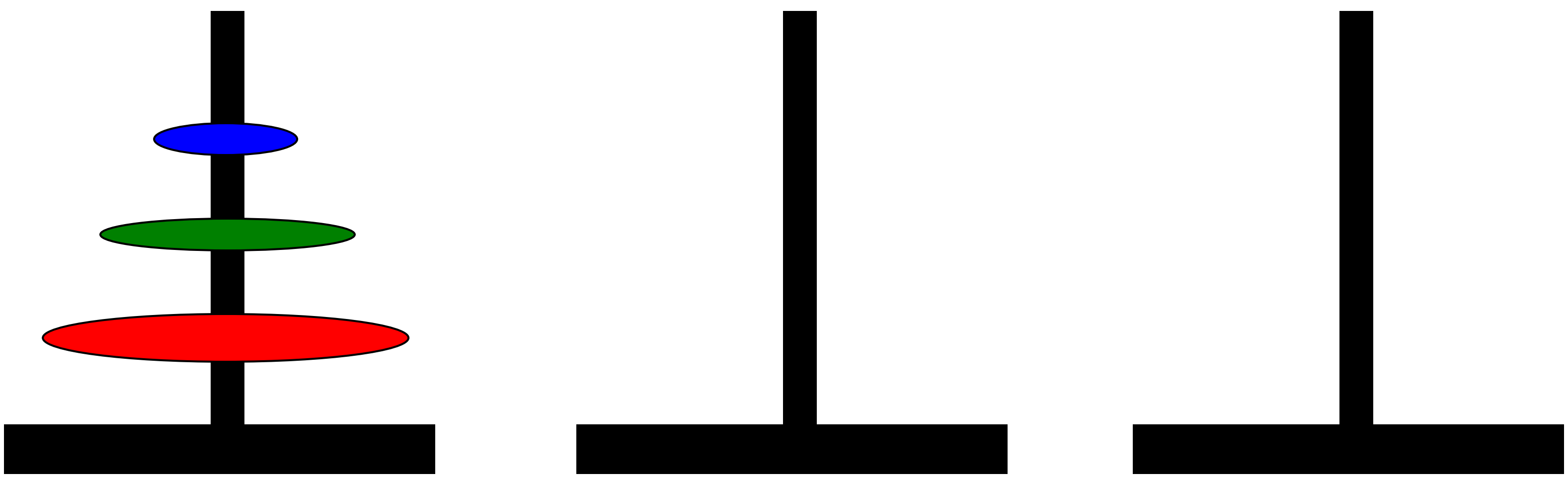
- There are three towers
- 64 gold disks, with decreasing sizes, placed on the first tower
- You need to move all of the disks from the first tower to the last tower
- Larger disks can not be placed on top of smaller disks
- The third tower can be used to temporarily hold disks

Tower of Hanoi

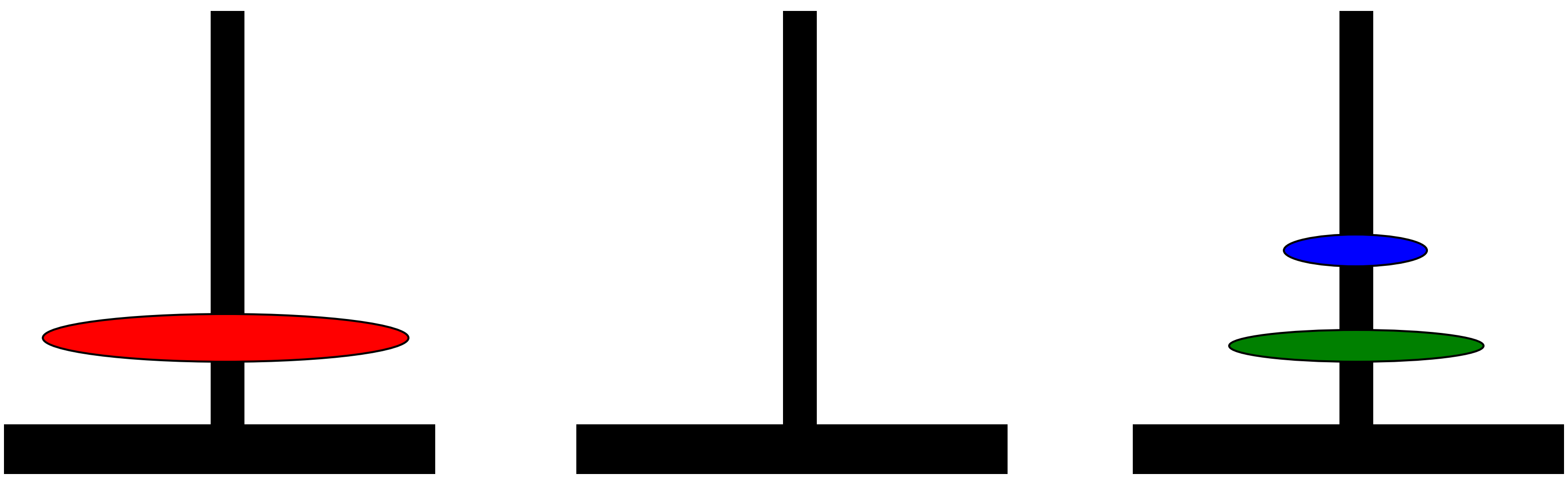
- The disks must be moved within one week. Assume one disk can be moved in 1 second. Is this possible?

- 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.

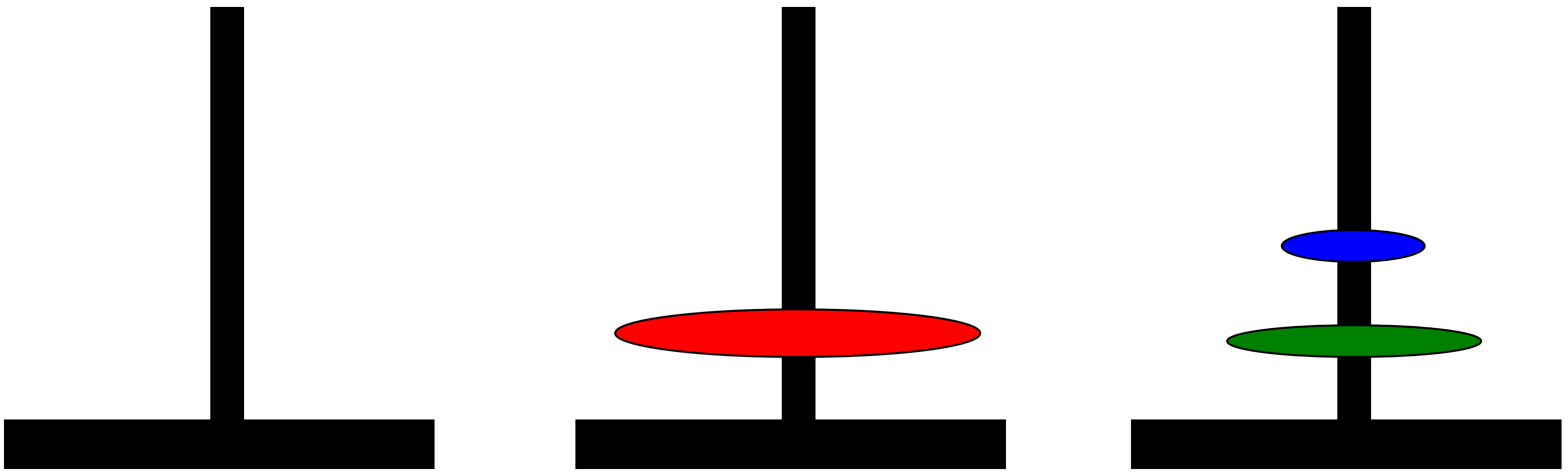
Recursive Solution



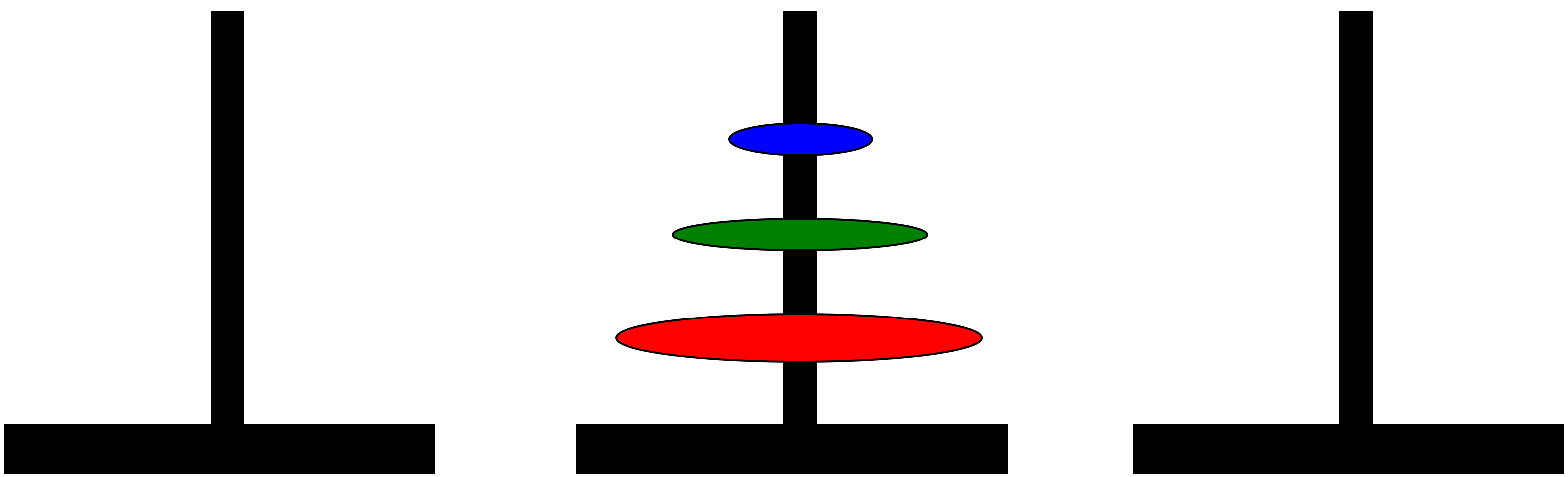
Recursive Solution



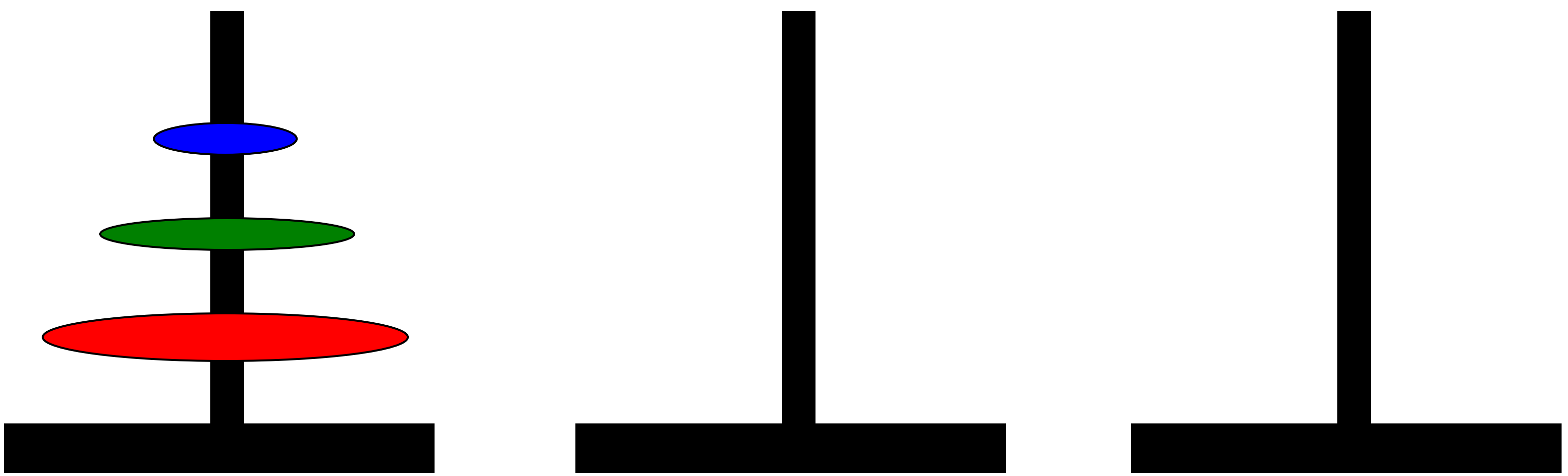
Recursive Solution



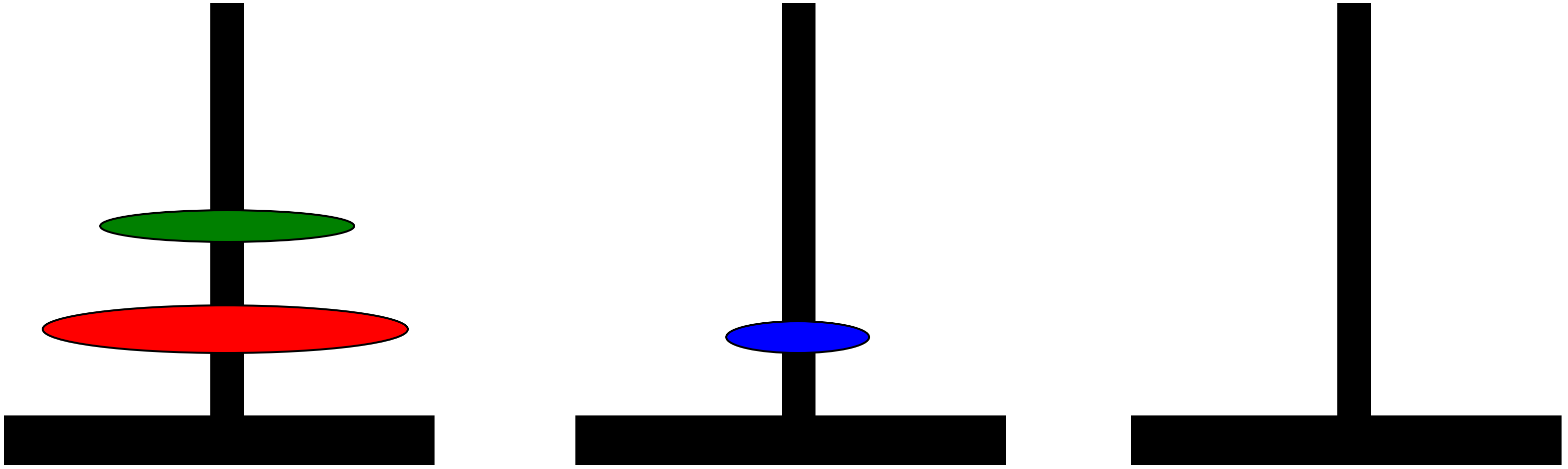
Recursive Solution



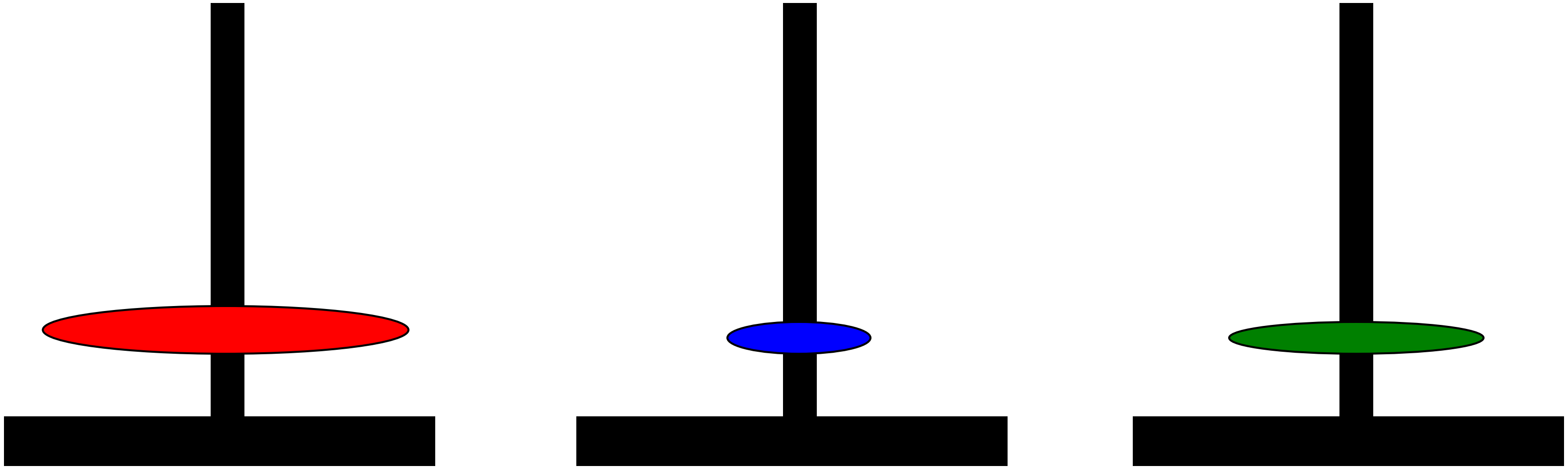
Tower of Hanoi



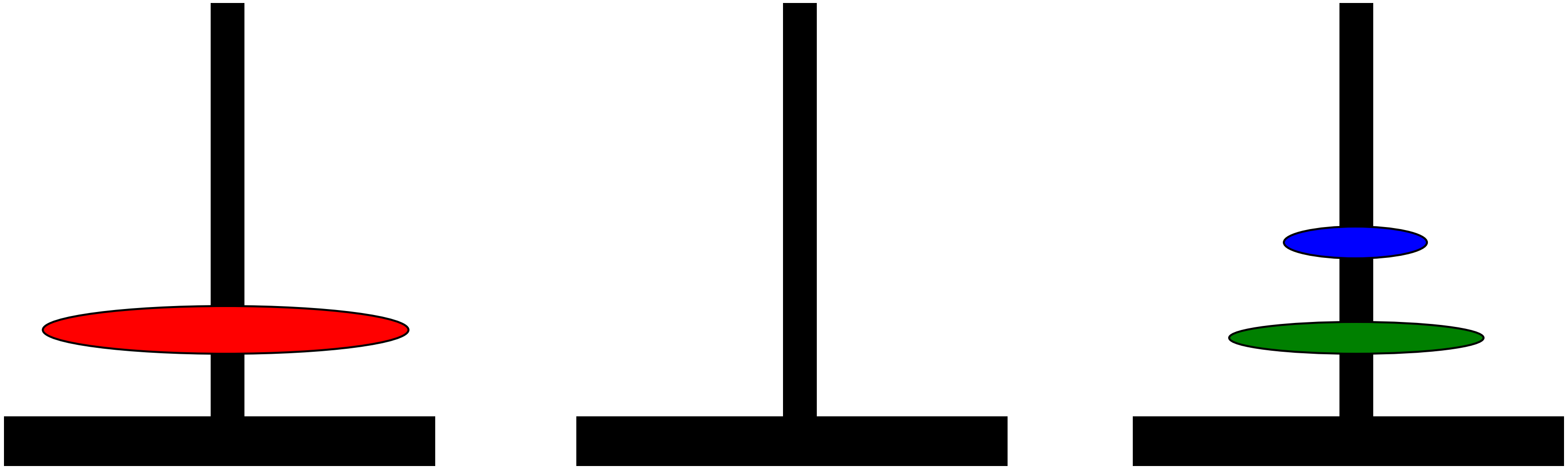
Tower of Hanoi



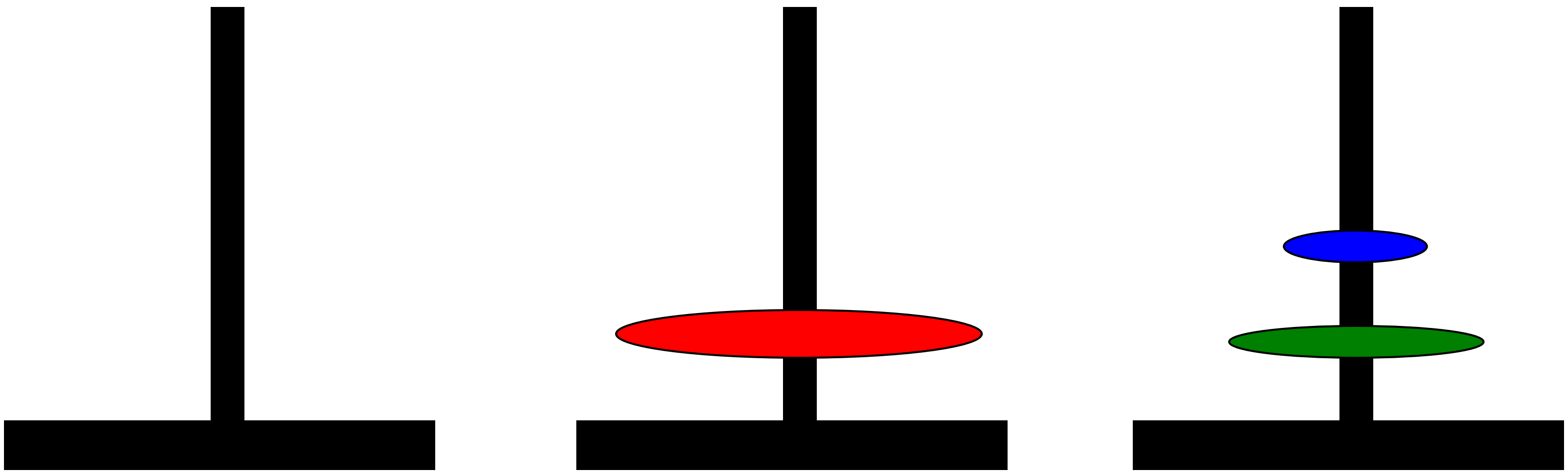
Tower of Hanoi



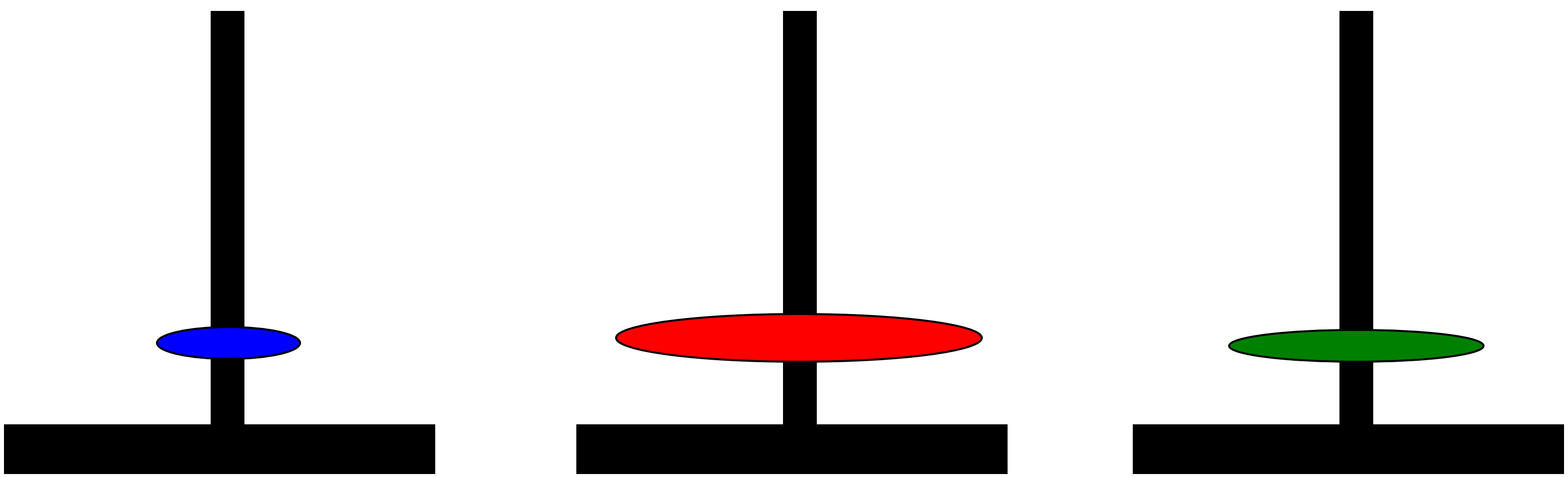
Tower of Hanoi



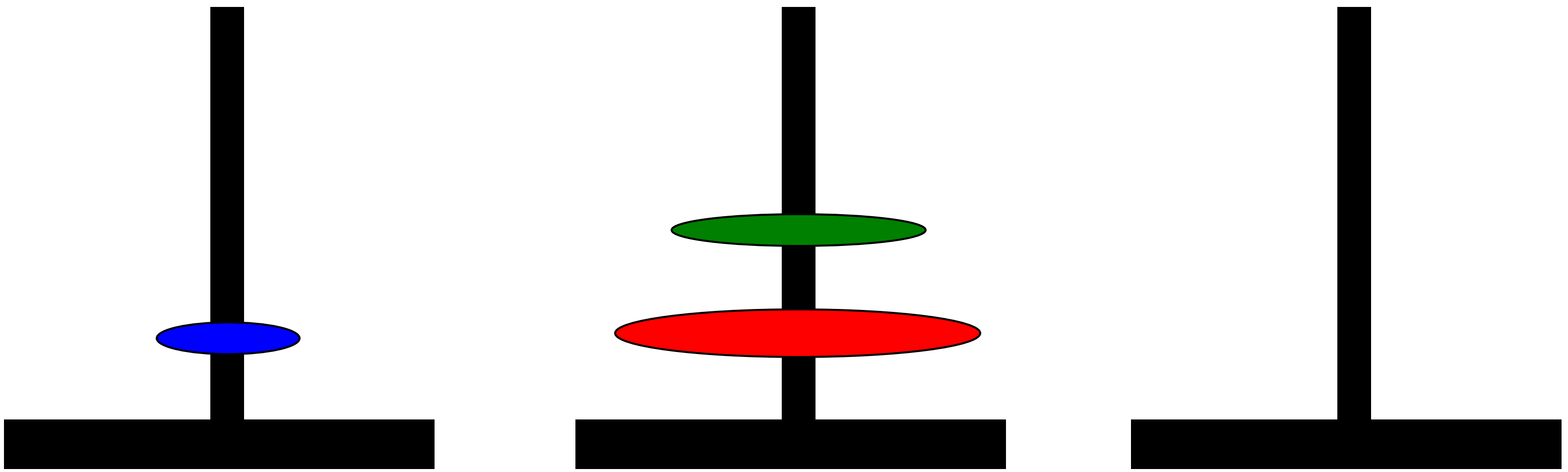
Tower of Hanoi



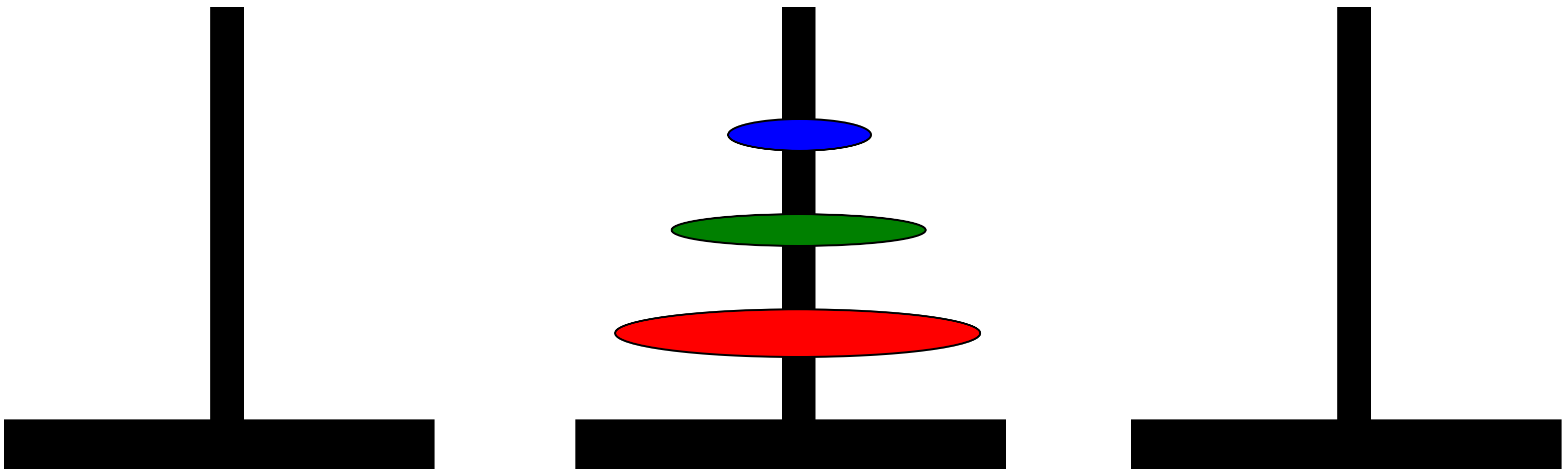
Tower of Hanoi



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);
    }
  }
```

Induction

- To prove a statement S(n) for positive integers n
  - Prove S(1)
  - Prove that if S(n) is true [inductive hypothesis] then S(n+1) is true.
- This implies that S(n) is true for n=1,2,3,…

Correctness and Cost

- Use induction to prove that the recursive algorithm solves the Tower of Hanoi problem.
- The number of moves M(n) required by the algorithm to solve the n-disk problem satisfies the recurrence relation
  - $M(n) = 2M(n-1) + 1$
  - $M(1) = 1$

Guess and Prove

- Calculate M(n) for small n and look for a pattern.
- Guess the result and prove your guess correct using induction.

n	M(n)
1	1
2	3
3	7
4	15
5	31

Substitution Method

- Unwind recurrence, by repeatedly replacing M(n) by the r.h.s. of the recurrence until the base case is encountered.

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

$= 2*[2*M(n-2)+1] + 1 = 2^2 * M(n-2) + 1+2$

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

$= 2^3 * M(n-3) + 1+2 + 2^2$

Geometric Series

- After k steps

$M(n) = 2^k * M(n\text{-}k) + 1+2 + 2^2 + \ldots + 2^{n-k-1}$

- Base case encountered when k = n-1

$M(n) = 2^{n-1} * M(1) + 1+2 + 2^2 + \ldots + 2^{n-2}$

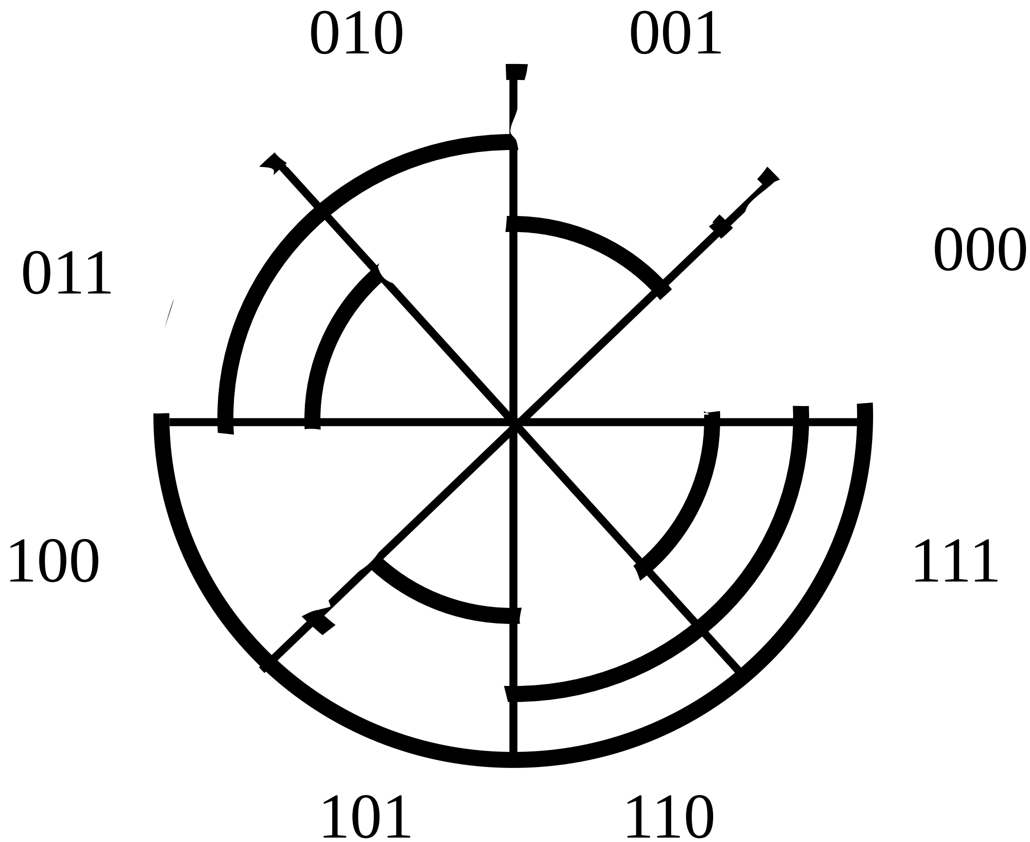
$$= 1 + 2 + \ldots + 2^{n-1} = \sum_{i=0}^{n-1} 2^i$$

- Use induction to reprove result for M(n) using this sum. Generalize by replacing 2 by x.

Gray Code

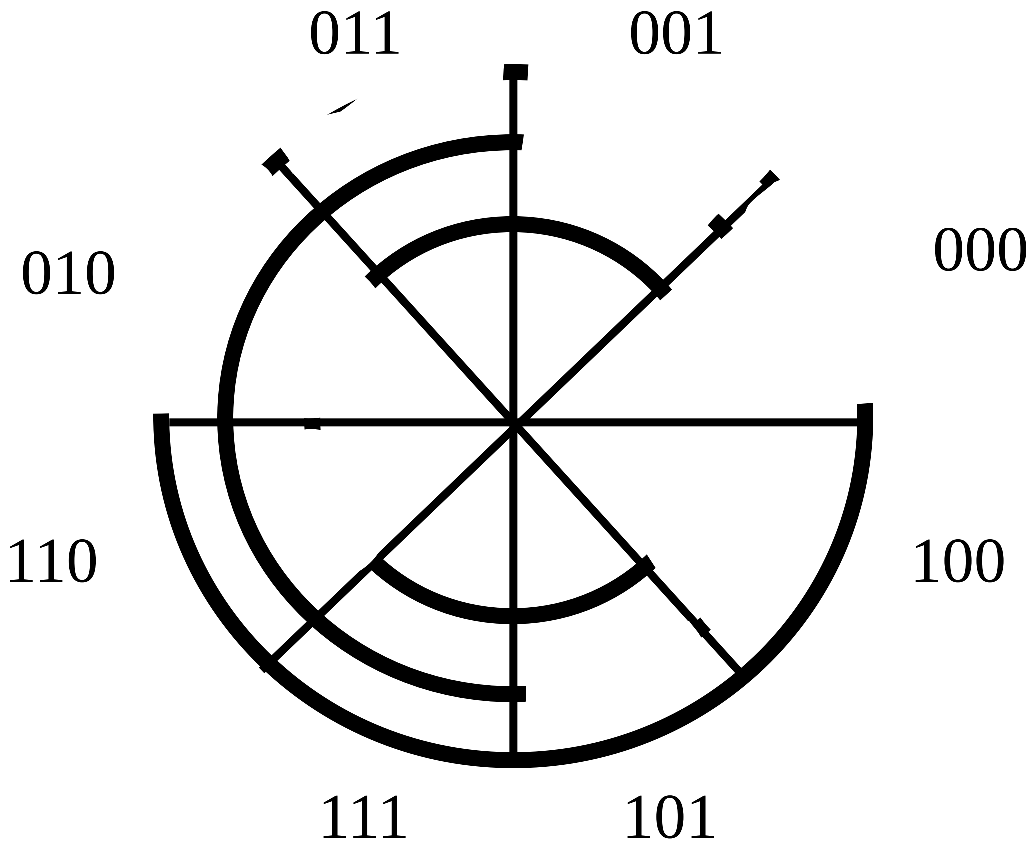
- An n-bit Gray code is a 1-1 onto mapping from $[0..2^n-1]$ such that the binary representation of consecutive numbers differ by exactly one bit.
- Invented by Frank Gray for a shaft encoder - a wheel with concentric strips and a conducting brush which can read the number of strips at a given angle. The idea is to encode 2^n different angles, each with a different number of strips, corresponding to the n-bit binary numbers.

Shaft Encoder (Counting Order)



Consecutive angles can have an abrupt change in the number of strips (bits) leading to potential detection errors.

Shaft Encoder (Gray Code)



Since a Gray code is used, consecutive angles have only one change in the number of strips (bits).

Binary-Reflected Gray Code

- $G_1 = [0,1]$
- $G_n = [0G_{n-1}, 1\overline{G}_{n-1}]$, $\overline{G} \Rightarrow$ reverse order $\equiv$ complement leading bit

- $G_2 = [0G_1, 1\overline{G}_1] = [00,01,11,10]$
- $G_3 = [0G_2, 1\overline{G}_2] =$ [000,001,011,010,110,111,101,100]
- Use induction to prove that this is a Gray code

Iterative Formula

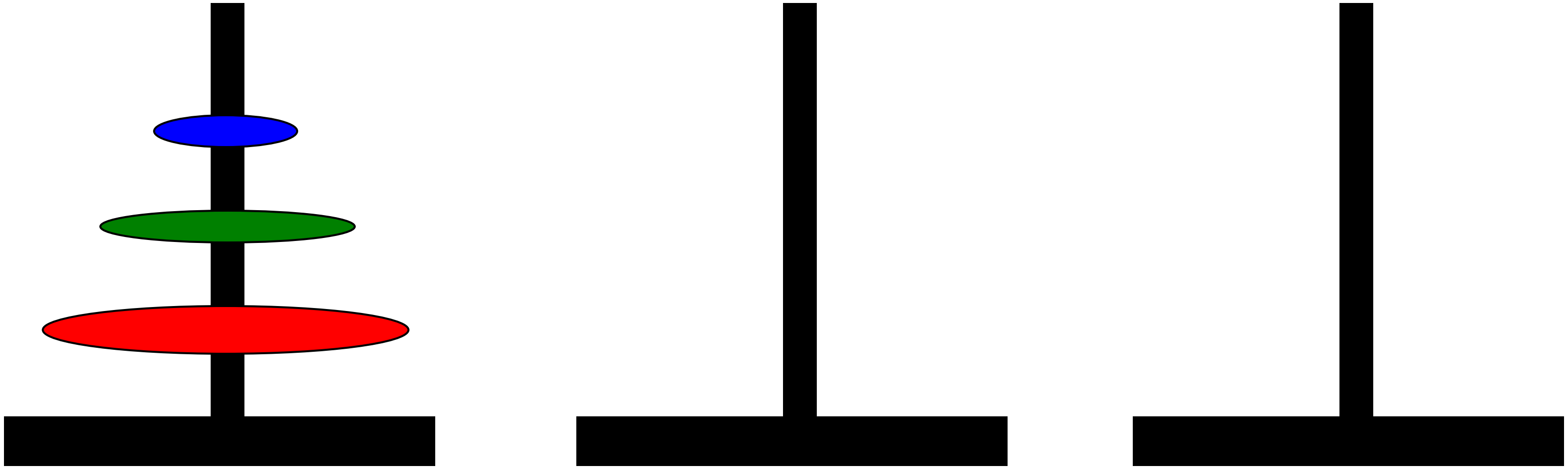
- Let $G_n(i)$ be a function from $[0,\ldots,2^n-1]$
- $G_n(i) = i \wedge (i >> 1)$ [exclusive or of i and i/2]
  - $G_2(0) = 0$, $G_2(1) = 1$, $G_2(2) = 3$, $G_2(3) = 2$
- Use induction to prove that the sequence $G_n(i)$, $i=0,\ldots,2^n-1$ is a binary-reflected Gray code.

Gray Code & Tower of Hanoi

- Introduce coordinates $(d_0,\ldots,d_{n-1})$, where $d_i \in \{0,1\}$
- Associate d_i with the ith disk
- Initialize to $(0,\ldots,0)$ and flip the ith coordinate when the i-th disk is moved
- The sequence of coordinate vectors obtained from the Tower of Hanoi solution is a Gray code (why?)

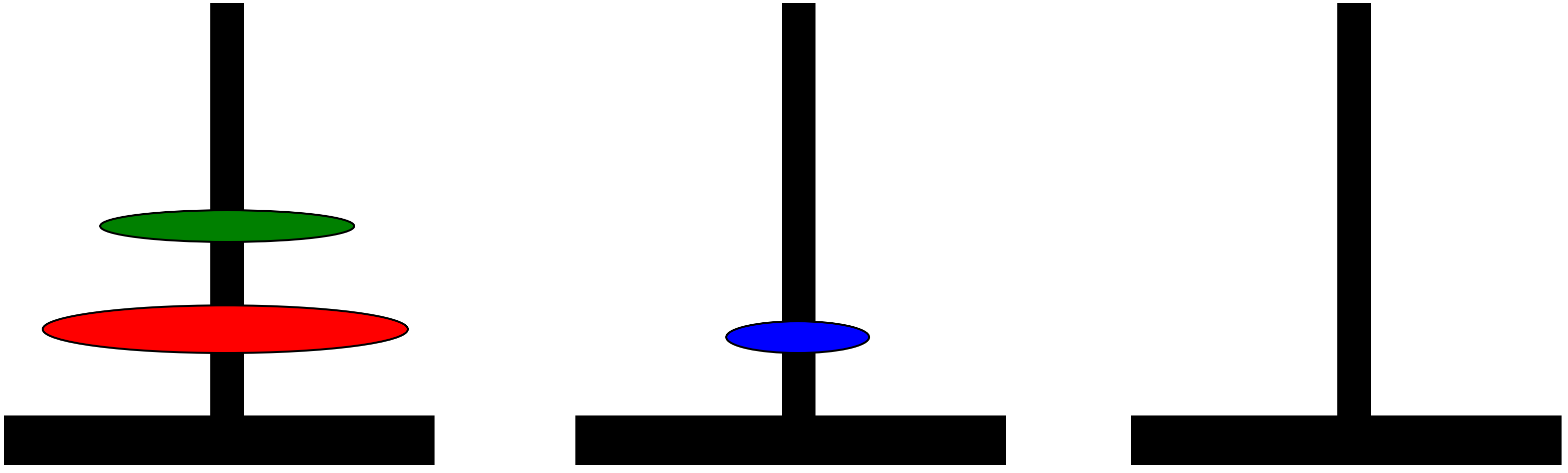
Tower of Hanoi

(0,0,0)



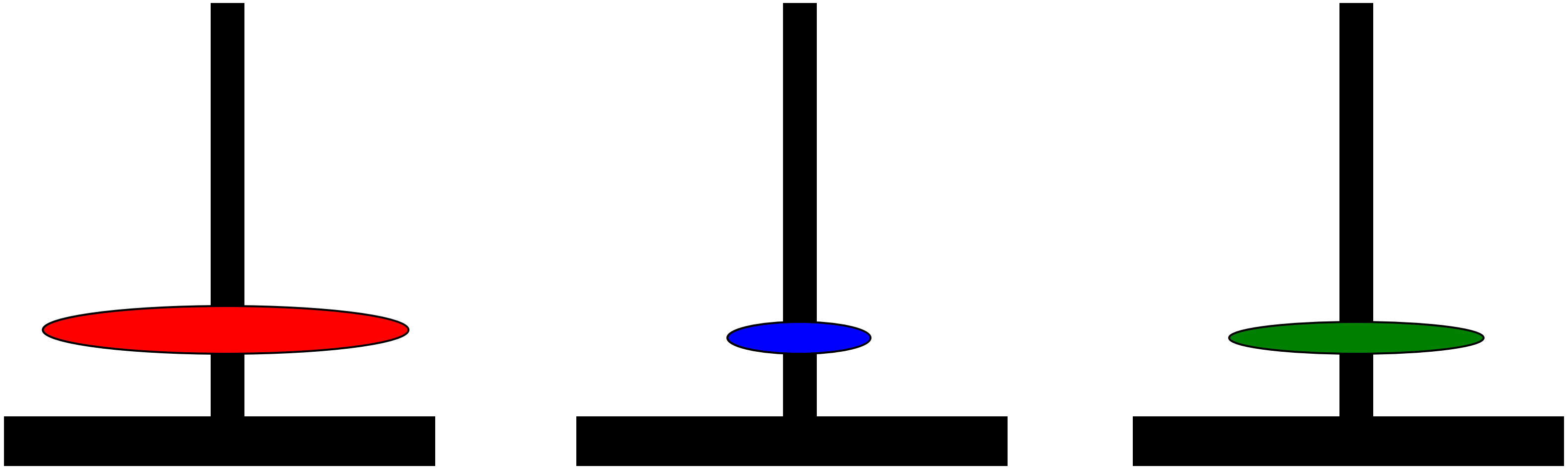
Tower of Hanoi

(0,0,1)



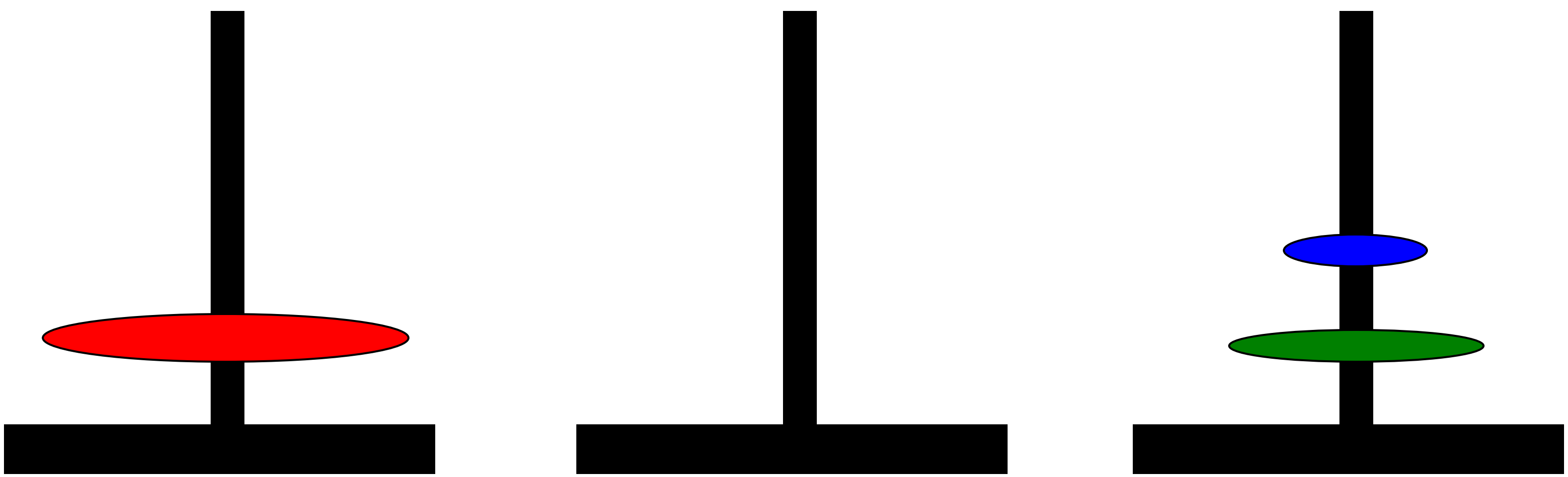
Tower of Hanoi

(0,1,1)



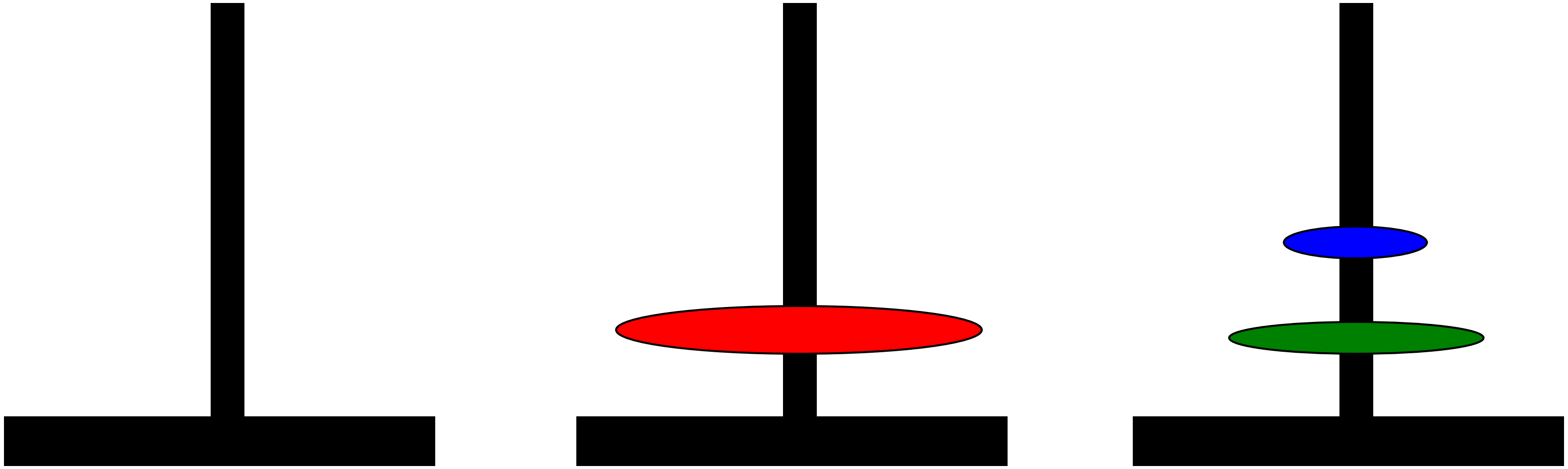
Tower of Hanoi

(0,1,0)



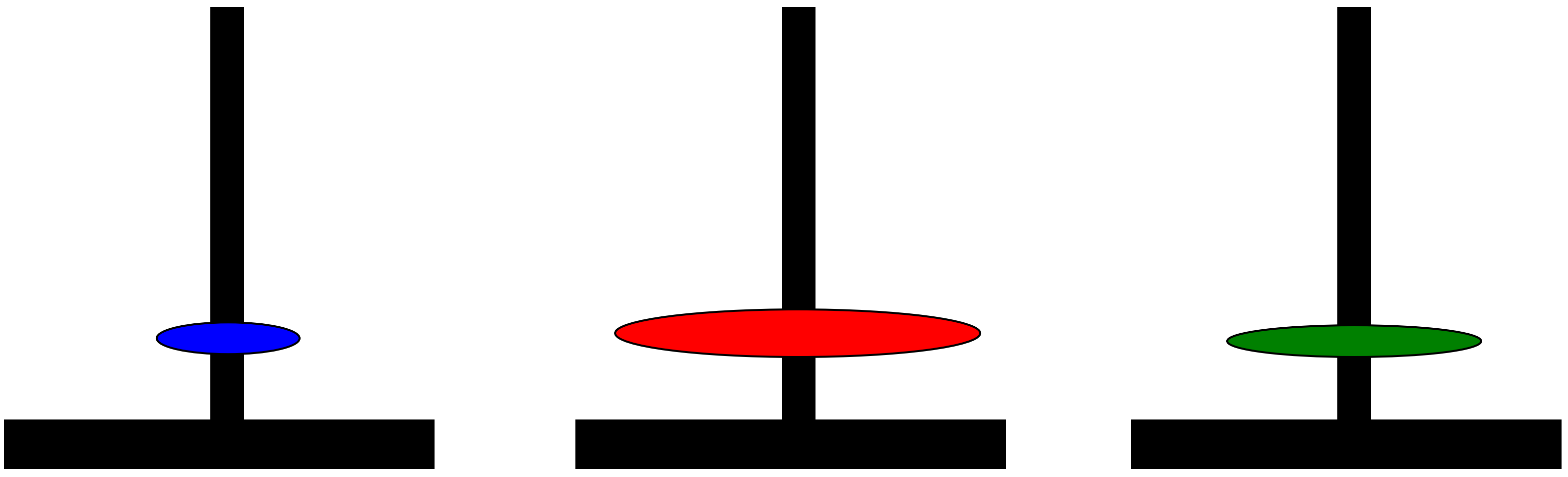
Tower of Hanoi

(1,1,0)



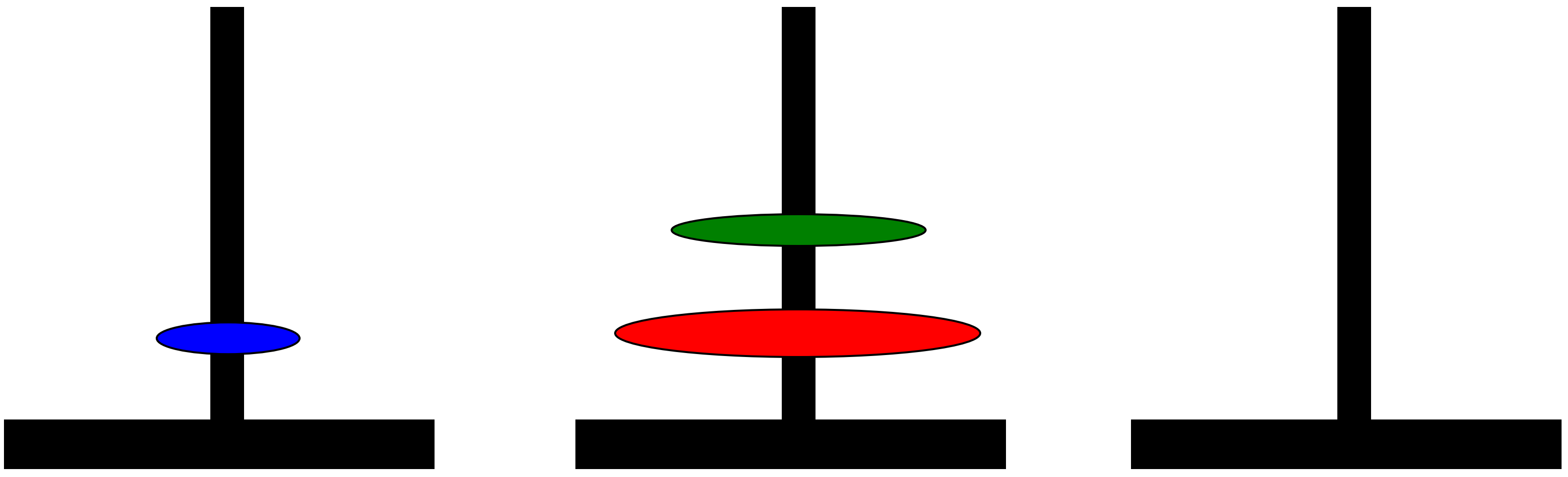
Tower of Hanoi

(1,1,1)



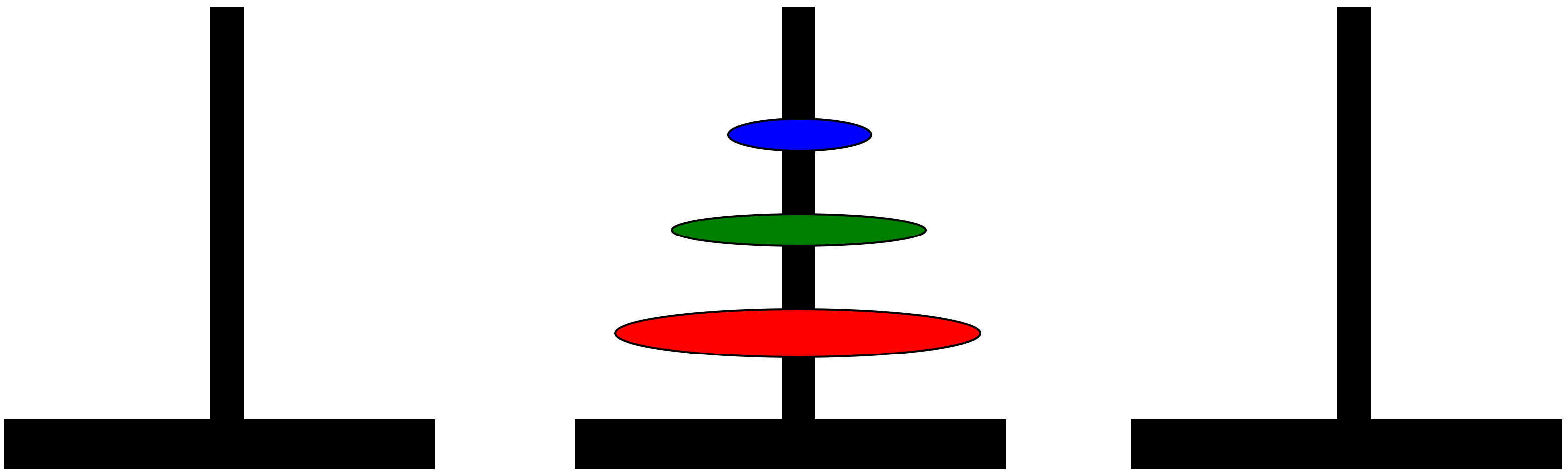
Tower of Hanoi

(1,0,1)



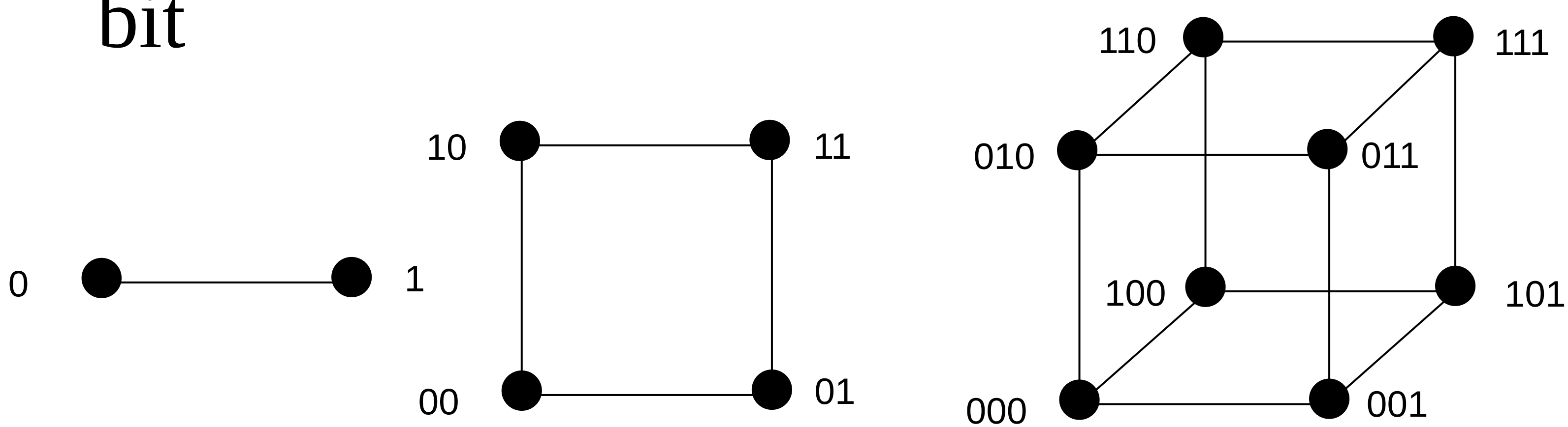
Tower of Hanoi

(1,0,0)



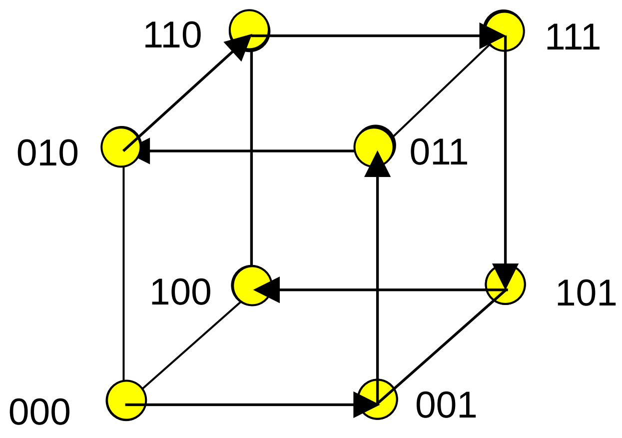
Hypercube

- Graph (recursively defined)
- n-dimensional cube has 2^n nodes with each node connected to n vertices
- Binary labels of adjacent nodes differ in one bit



Hypercube, Gray Code and Tower of Hanoi

- A Hamiltonian path is a sequence of edges that visit each node exactly once.
- A Hamiltonian path on a hypercube provides a Gray code (why?)



Hypercube and Gray Code