

DATA STRUCTURES AND ALGORITHMS

CSE220

Prof. Ramesh Ragala

July 22, 2015

- Algorithm for summation of N numbers

Algorithm Sum(a, n)

```
1: {  
2:  $sum \leftarrow 0$ ;  
3: for  $i \leftarrow 1$  to  $n$  do  
4:    $sum \leftarrow sum + a[i]$ ;  
5: end for  
6: }
```

- Algorithm for Finding Largest Number in a given set

Algorithm Max(a,n)

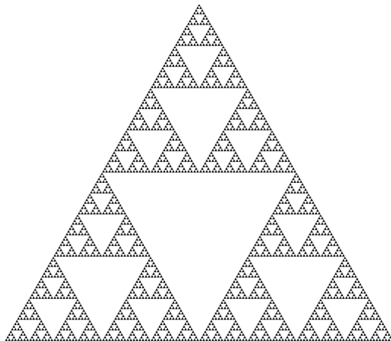
```
1: {  
2: larger  $\leftarrow$  a[0];  
3: for i  $\leftarrow$  1 to n do  
4:   if (a[i] > larger) then  
5:     larger  $\leftarrow$  a[i];  
6:   end if  
7: end for  
8: return larger  
9: }
```

- Algorithm for Matrix Addition

Algorithm MatrixAdd(a, b, n, m)

```
1: {  
2:  for  $i \leftarrow 1$  to  $n$  do  
3:    for  $j \leftarrow 1$  to  $m$  do  
4:       $c[i][j] \leftarrow a[i][j] + b[i][j];$   
5:    end for  
6:  end for  
7: }
```

- Analyse the process of drawing the following picture



- An algorithm is said to be recursive if the same algorithm is invoked in the body of the algorithm
- It solves the problem by possibly using the result of applying itself to a simpler problem
- Properties of Recursive Calls
 - It solves the large problem by using its solution to a simpler sub-problem.
 - Example: Divide and Conquer Approach
 - Eventually the sub-problem is simple enough that it can be solved without applying the algorithm to it recursively. **This is called Base Case**

- **Base Case**

- There should be at least one base case in recursive algorithm.
- It is used to avoid the infinite looping of Recursive Calls.
- Every possible chain of recursive calls must eventually reach a base case.

- **Recursive Call**

- Calls to the current method
- Each recursive call should be defined so that it makes progress towards a base case.

- **Two types of Recursive Algorithm**

- Direct Recursive Algorithm:
 - An algorithm that, directly calls it self is called direct recursive
- Indirect Recursive Algorithm:
 - An algorithm - A said to be indirect recursive, if it calls another algorithm, which in turn calls Algorithm - A

- Recursive Algorithm for Finding Largest Number in a given set

Algorithm $\text{RecurMax}(a,n)$

```
1: {  
2: if ( $n = 1$ ) then  
3:   return  $a[0]$ ;  
4: else  
5:    $\text{max}\{ \text{RecurMax}(a,n-1), a(n-1) \}$ ;  
6: end if  
7: }
```

- Tower of Hanoi is a mathematical puzzle invented by a French Mathematician **Edouard Lucas** in 1883.
- The game starts by having few discs **stacked in increasing order of size**.
- The number of discs can vary, but there are **only** three Towers.
- The goal of Towers of Hanoi is to move all the disks from the leftmost Tower to the rightmost Tower, adhering to the following rules:
 - ① Move only one disk at a time.

- Tower of Hanoi is a mathematical puzzle invented by a French Mathematician **Edouard Lucas** in 1883.
- The game starts by having few discs **stacked in increasing order of size**.
- The number of discs can vary, but there are **only** three Towers.
- The goal of Towers of Hanoi is to move all the disks from the leftmost Tower to the rightmost Tower, adhering to the following rules:
 - 1 Move only one disk at a time.
 - 2 A larger disk may not be placed on top of a smaller disk.

- Tower of Hanoi is a mathematical puzzle invented by a French Mathematician **Edouard Lucas** in 1883.
- The game starts by having few discs **stacked in increasing order of size**.
- The number of discs can vary, but there are **only** three Towers.
- The goal of Towers of Hanoi is to move all the disks from the leftmost Tower to the rightmost Tower, adhering to the following rules:
 - ① Move only one disk at a time.
 - ② A larger disk may not be placed on top of a smaller disk.
 - ③ All disks, except the one being moved, must be on a Tower.

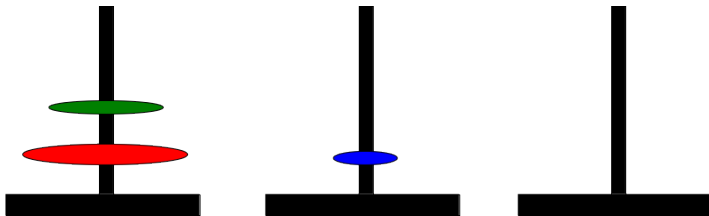
- Towers of Hanoi Problem : Initial Position



- Towers of Hanoi Problem : Step - 1

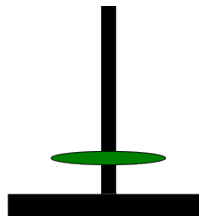
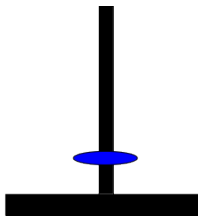
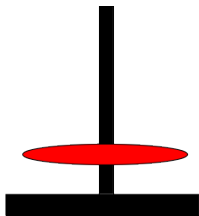


- Towers of Hanoi Problem : Step - 2

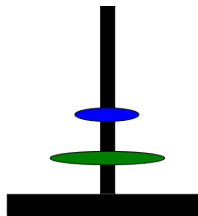
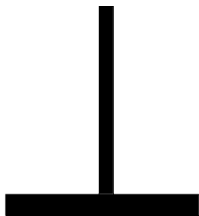
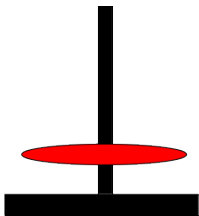


RECURSIVE ALGORITHM EXAMPLE

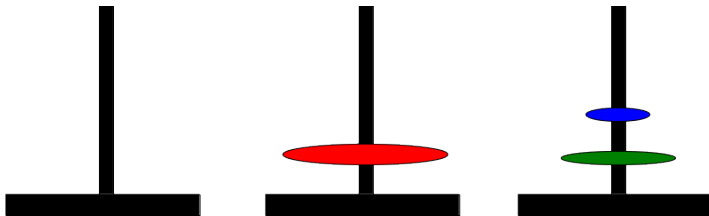
- Towers of Hanoi Problem : Step - 3



- Towers of Hanoi Problem : Step - 4

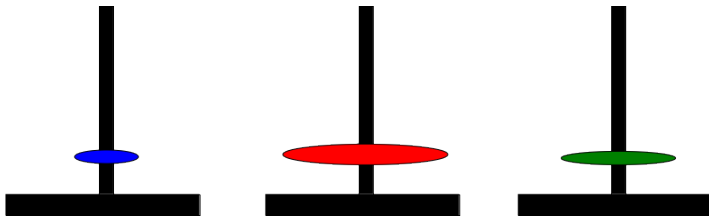


- Towers of Hanoi Problem : Step - 5

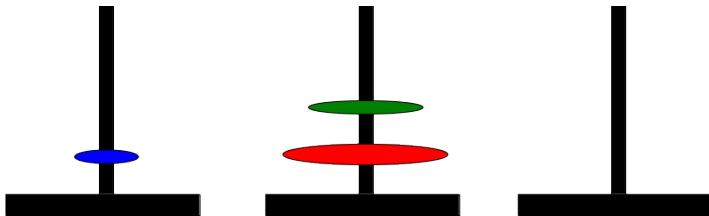


RECURSIVE ALGORITHM EXAMPLE

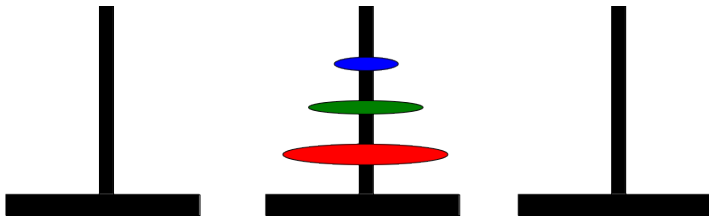
- Towers of Hanoi Problem : Step - 6



- Towers of Hanoi Problem : Step - 7



- Towers of Hanoi Problem : Step - 8



- Recursive Algorithm for Towers of Hanoi

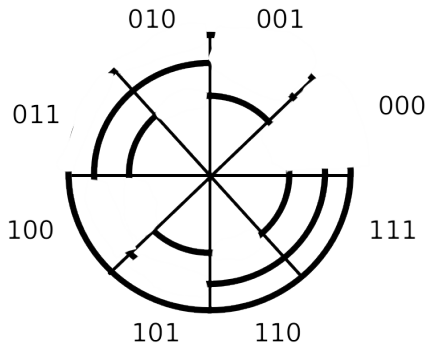
Algorithm ToH(n, x, y, z)

```
1: {  
2: if ( $n \geq 1$ ) then  
3:   ToH( $n-1, x, z, y$ );  
4:   Write(" Move to disk from tower" ,  $x$ , " to top of tower" ,  $y$ );  
5:   ToH( $n-1, z, y, x$ );  
6: end if  
7: }
```

- **Gray Code**

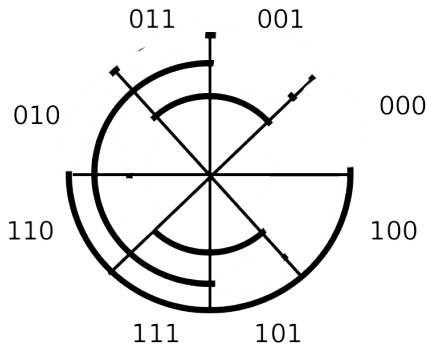
- An n-bit Gray code is a 1-1 onto mapping from $[0..2^n-1]$ such that the binary representation of consecutive numbers differ by exactly one bit.
- Invented by **Frank Gray** for a shaft encoder → it is a wheel with concentric strips and a conducting brush which can read the number of strips at a given 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).

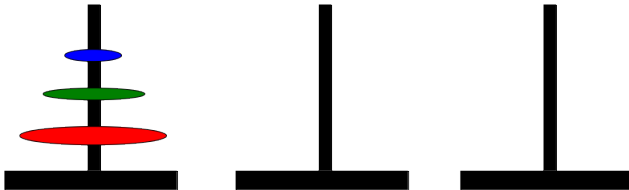


- **Gray Code and Towers of Hanoi**

- Assume the coordinates (d_0, \dots, d_{n-1}) , where $d_i \in \{0, 1\}$
- Associate d_i with the i^{th} disk
- Initialize to $(0, \dots, 0)$ and flip the i^{th} 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??**

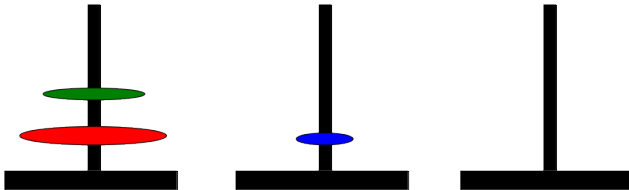
- Gray Code and Towers of Hanoi Problem

$(0,0,0)$



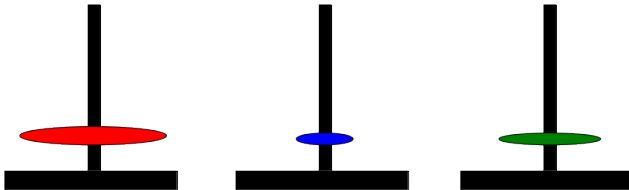
- Gray Code and Towers of Hanoi Problem

$(0,0,1)$



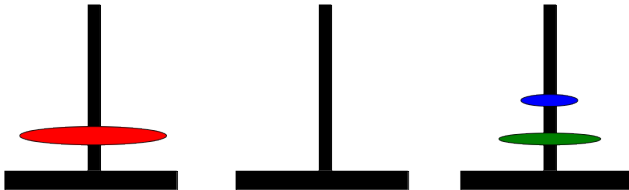
- Gray Code and Towers of Hanoi Problem

$(0,1,1)$



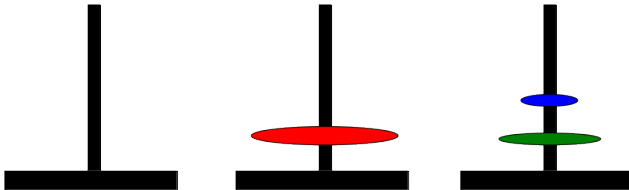
- Gray Code and Towers of Hanoi Problem

$(0,1,0)$



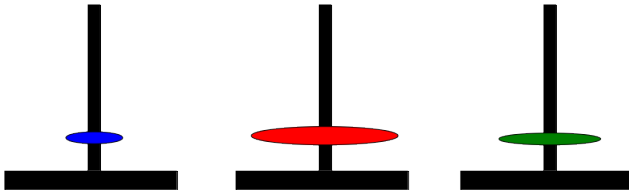
- Gray Code and Towers of Hanoi Problem

$(1,1,0)$



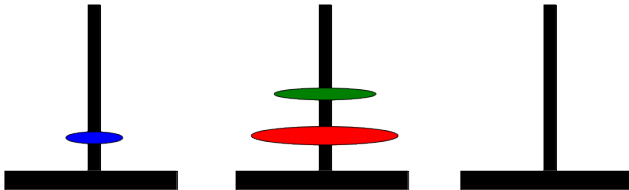
- Gray Code and Towers of Hanoi Problem

(1,1,1)



- Gray Code and Towers of Hanoi Problem

$(1,0,1)$



- Gray Code and Towers of Hanoi Problem

$(1,0,0)$

