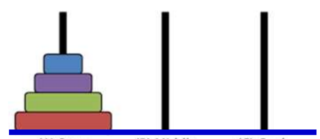
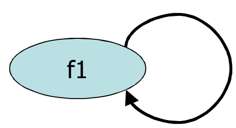
CSE 2215:
Data Structures and Algorithms-I

Recursion
Towers of Hanoi



Recursion

- The process in which a function calls itself is called recursion, and the corresponding function is called as recursive function.



Why Recursion?

- Recursion is a very useful and powerful technique.
- Using recursive algorithm, certain problems can be solved quite easily. Examples of such problems are
  - Towers of Hanoi (TOH),
  - Inorder/Preorder/Postorder Tree Traversals,
  - DFS of Graph, etc.

Recursion

How a problem is solved using recursion?

- In the recursive program,
  - the solution to the base case is provided [basis step], and
  - the solution of the bigger problem is expressed in terms of smaller problems [recursive step].

if this is a simple case
solve it
else
redefine the problem using recursion

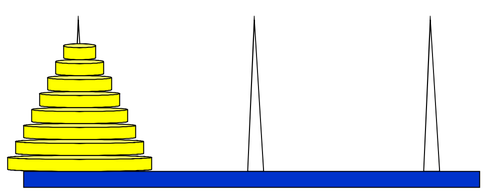
Why Stack Overflow error occurs in recursion?

- If the base case is not reached or not defined, then the stack overflow problem may arise.

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi Problem

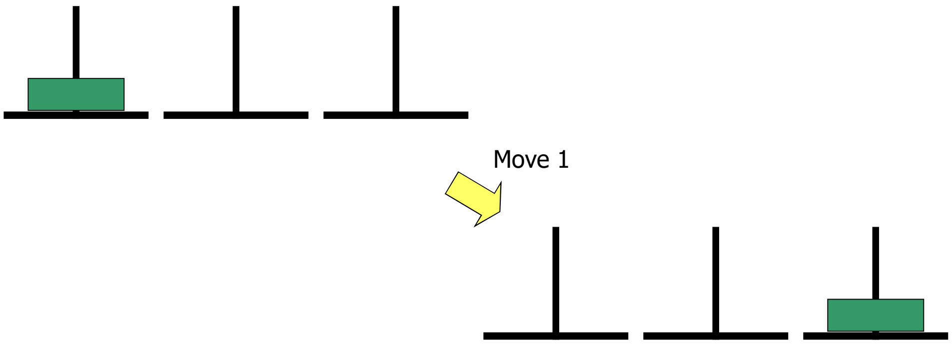
- The towers of Hanoi problem involves moving a number of disks (in different sizes) from one tower (or called "peg") to another.
  - The constraint is that the larger disk can never be placed on top of a smaller disk.
  - Only one disk can be moved at each time
  - Assume there are three towers available



Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi

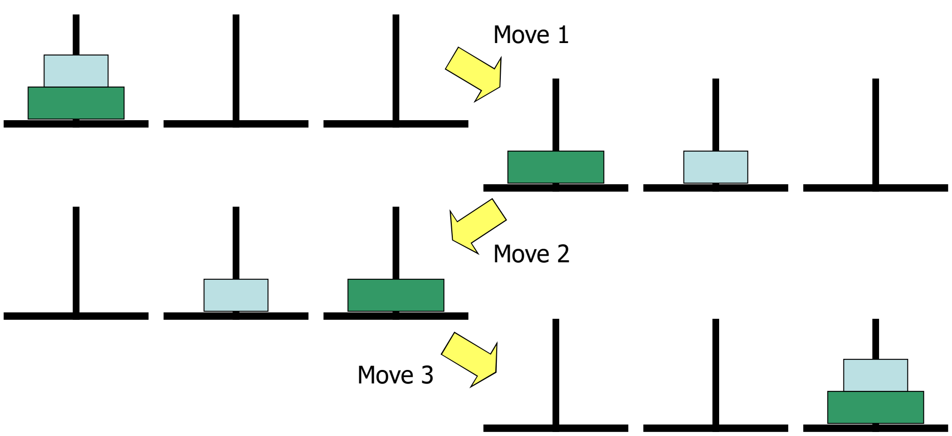
It is very easy if we only have 1 disk.



A total of 1 step only.

Towers of Hanoi

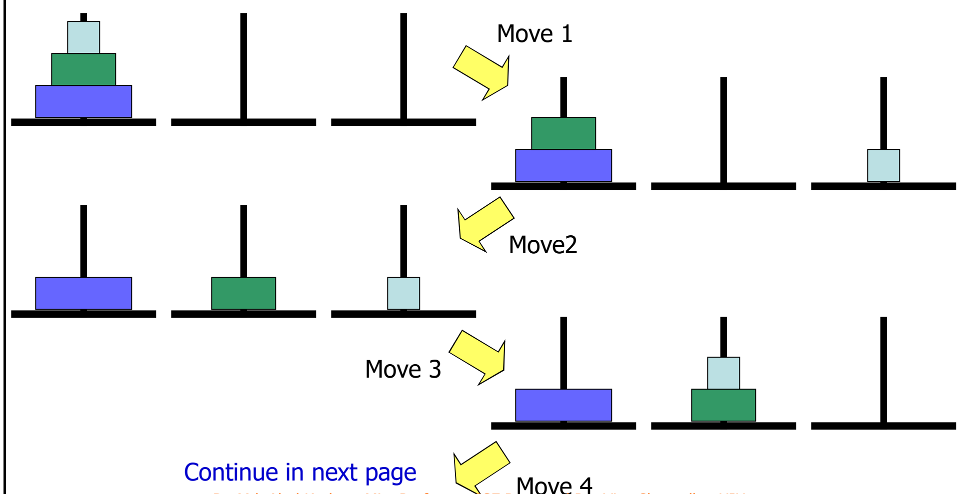
It is easy if we only have 2 disks.



A total of 3 steps.

Towers of Hanoi

It is not difficult if we only have 3 disks.

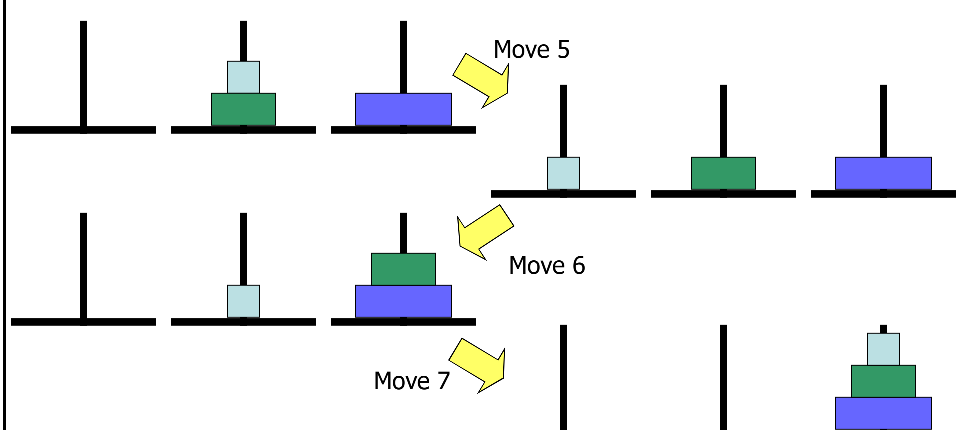


Continue in next page

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi

It is not difficult if we only have 3 disks.

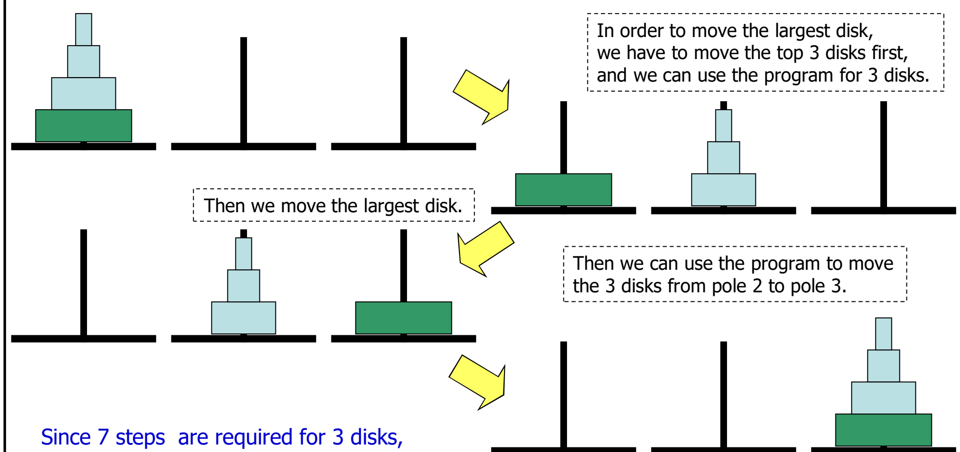


A total of 7 steps.

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi

Can you use it to solve for 4 disks?

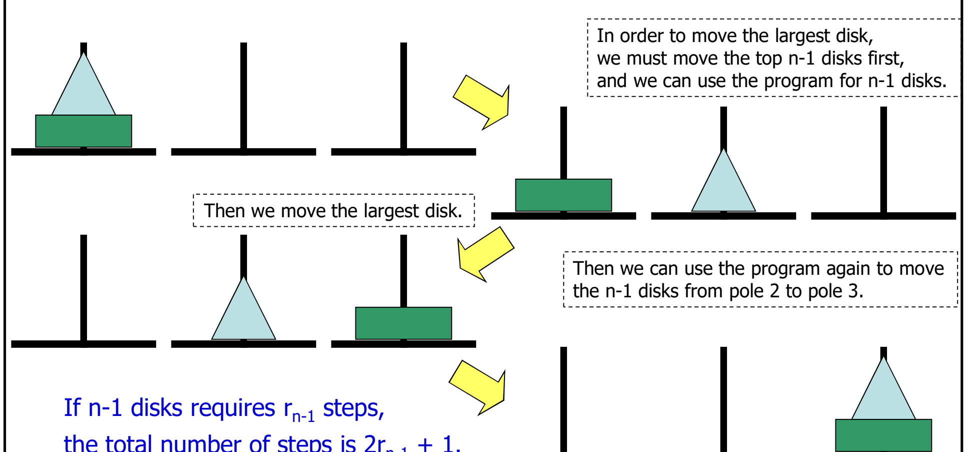


Since 7 steps are required for 3 disks, the total number of steps for 4 disks is 15.

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi

This recursion is true for any n.



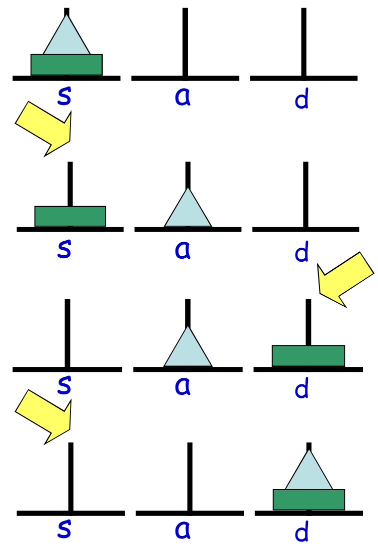
If n-1 disks requires r_{n-1} steps, the total number of steps is $2r_{n-1} + 1$.

$$\boxed{r_n = 2r_{n-1} + 1}$$

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi: Recursive Algorithm

```
void Hanoi(int n, int s, int a, int d) {
    if (n==1)
        print("%d —> %d", s, d);
    else {
        Hanoi(int n-1, int s, int d, int a);
        print("%d —> %d", s, d);
        Hanoi(int n-1, int a, int s, int d);
    }
}
```

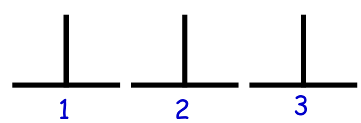


Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi: Recursion Tree

void Hanoi(int n, int s, int a, int d)

T(int n, int 1, int 2, int 3)



T(3, 1, 2, 3)

T(2, 1, 3, 2) | 1 —> 3 [4] | T(2, 2, 1, 3)

T(1, 1, 2, 3) → 1 —> 3 [1]

1 —> 2 [2]

T(1, 3, 1, 2) → 3 —> 2 [3]

T(1, 2, 3, 1) → 2 —> 1 [5]

2 —> 3 [6]

T(1, 1, 2, 3) → 1 —> 3 [7]

Output (Sequence of Disk Movement):

1 —> 3 1 —> 2 3 —> 2 1 —> 3 2 —> 1 2 —> 3 1 —> 3

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Towers of Hanoi: No. of Movements

No. of Disk Movements is $a_k = 2a_{k-1} + 1$

$a_1 = 1 = 2^1 - 1$

$a_2 = 2a_1 + 1 = 3 = 2^2 - 1$

$a_3 = 2a_2 + 1 = 2*3 + 1 = 7 = 2^3 - 1$

$a_4 = 2a_3 + 1 = 2*7 + 1 = 15 = 2^4 - 1$

$a_5 = 2a_4 + 1 = 2*15 + 1 = 31 = 2^5 - 1$

$a_6 = 2a_5 + 1 = 2*31 + 1 = 63 = 2^6 - 1$

By guessing the pattern: $a_k = 2^k-1$

You can verify by induction.