COMP 352

Tutorial Session 3

Session Outline

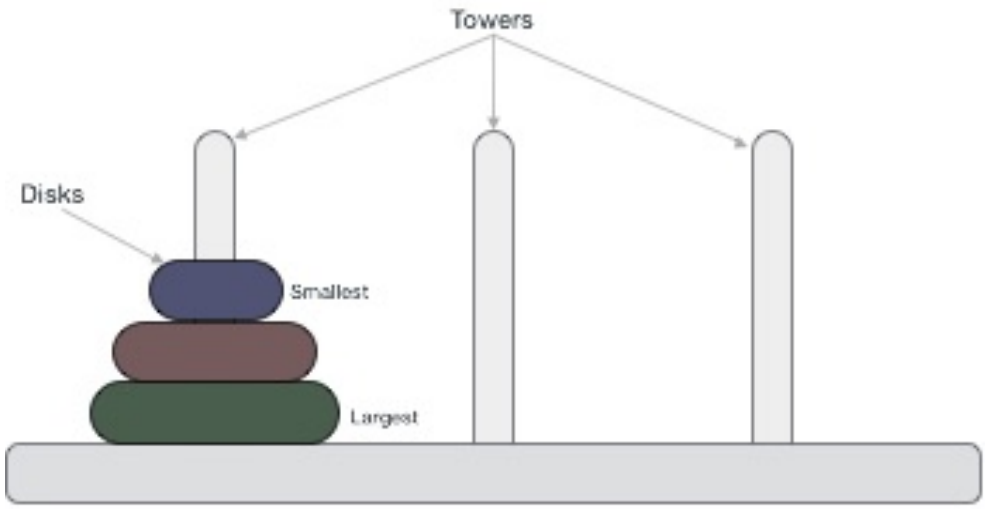
- Recursion - Tower of Hanoi
- Exercises:
  - Big-Oh complexity calculation
  - Recursion exercises
  - Proof of Big-Oh complexity

Recursion - Tower of Hanoi

- Rules:
- Only one disk can be moved among the towers at any given time.
- Only the "top" disk can be removed.
- No large disk can sit over a small disk.

- Demo example

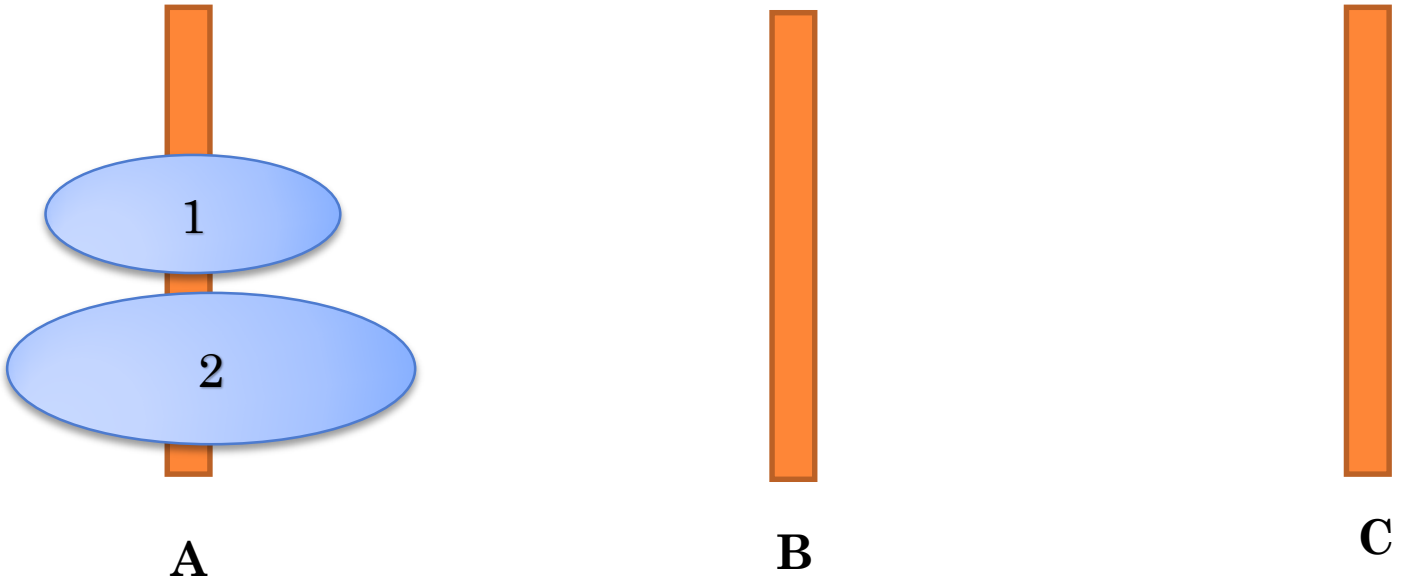
https://www.youtube.com/watch?v=YstLjLCGmgg



Reference: https://www.tutorialspoint.com/data_structures_algorithms/tower_of_hanoi.htm

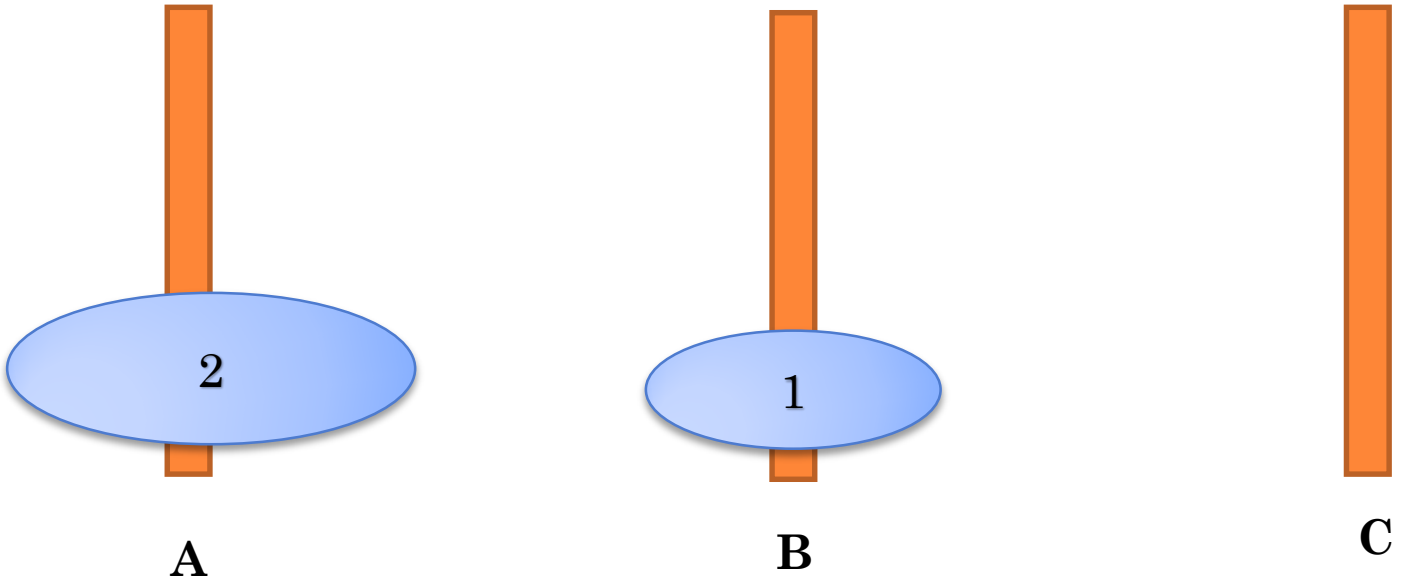
Illustrative Example 1

Move two disks from A to C



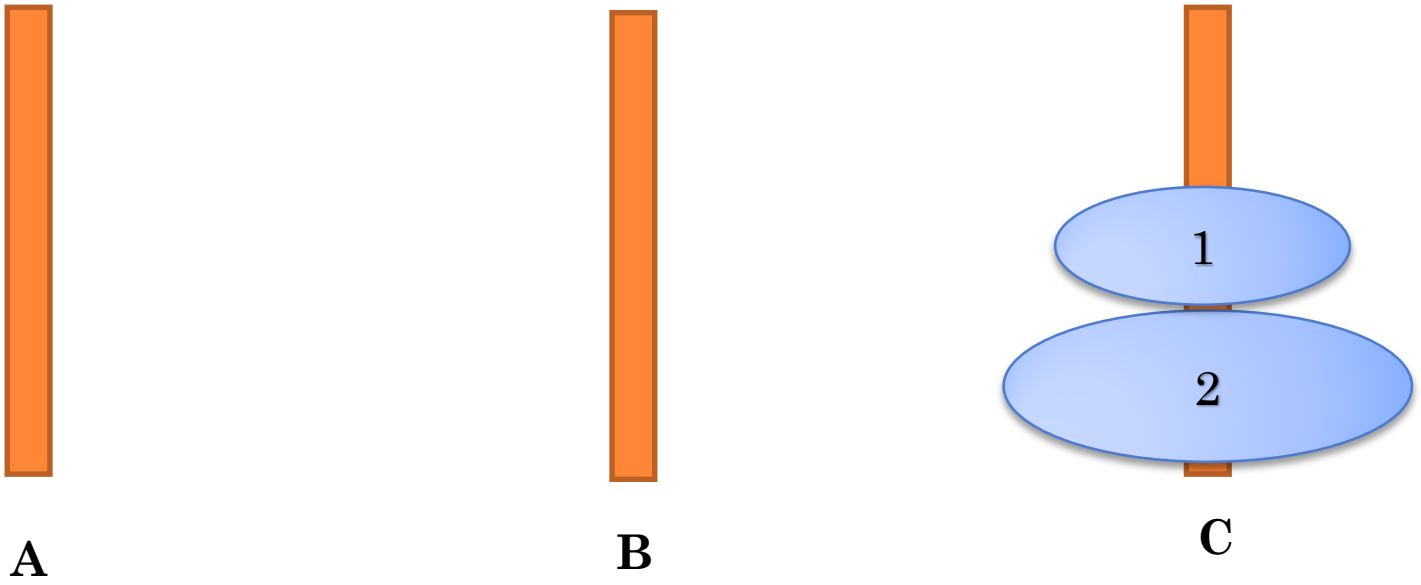
Illustrative Example 1

Move disk 1 from A to B



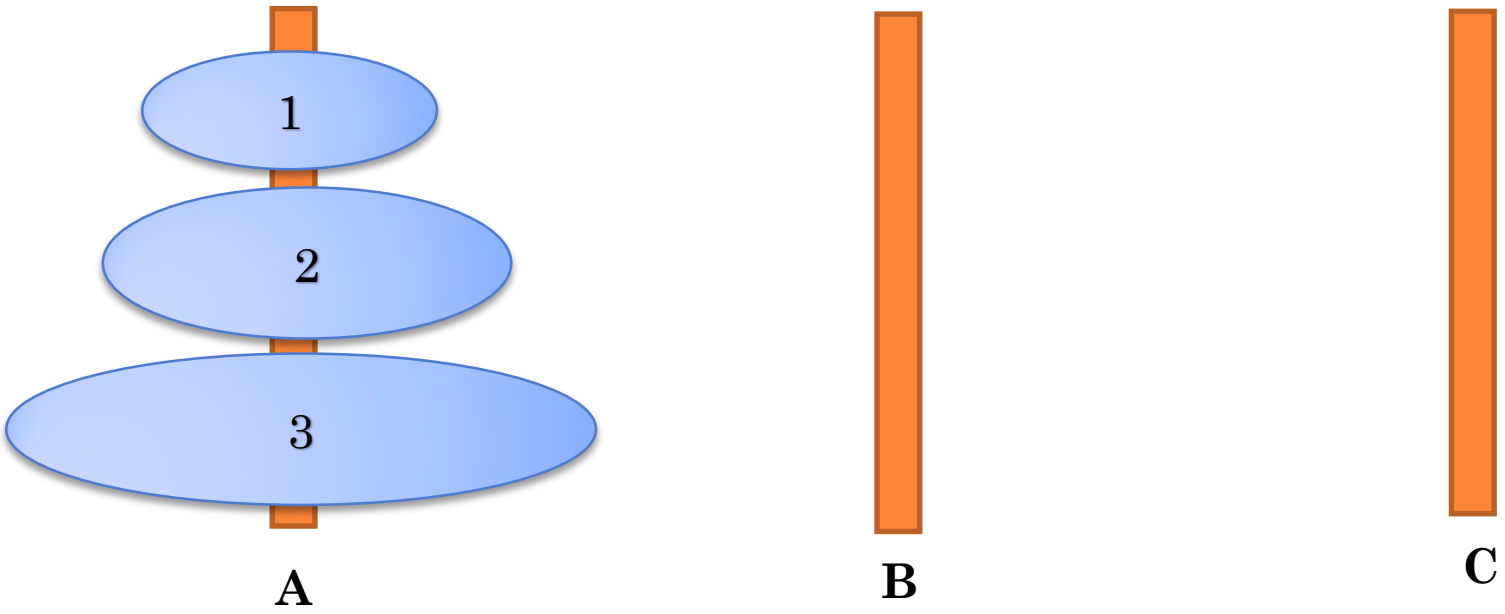
Illustrative Example 1

Move disk 2 from A to c
Move disk 1 from B to C



Illustrative Example 2

Move three disks from A to C

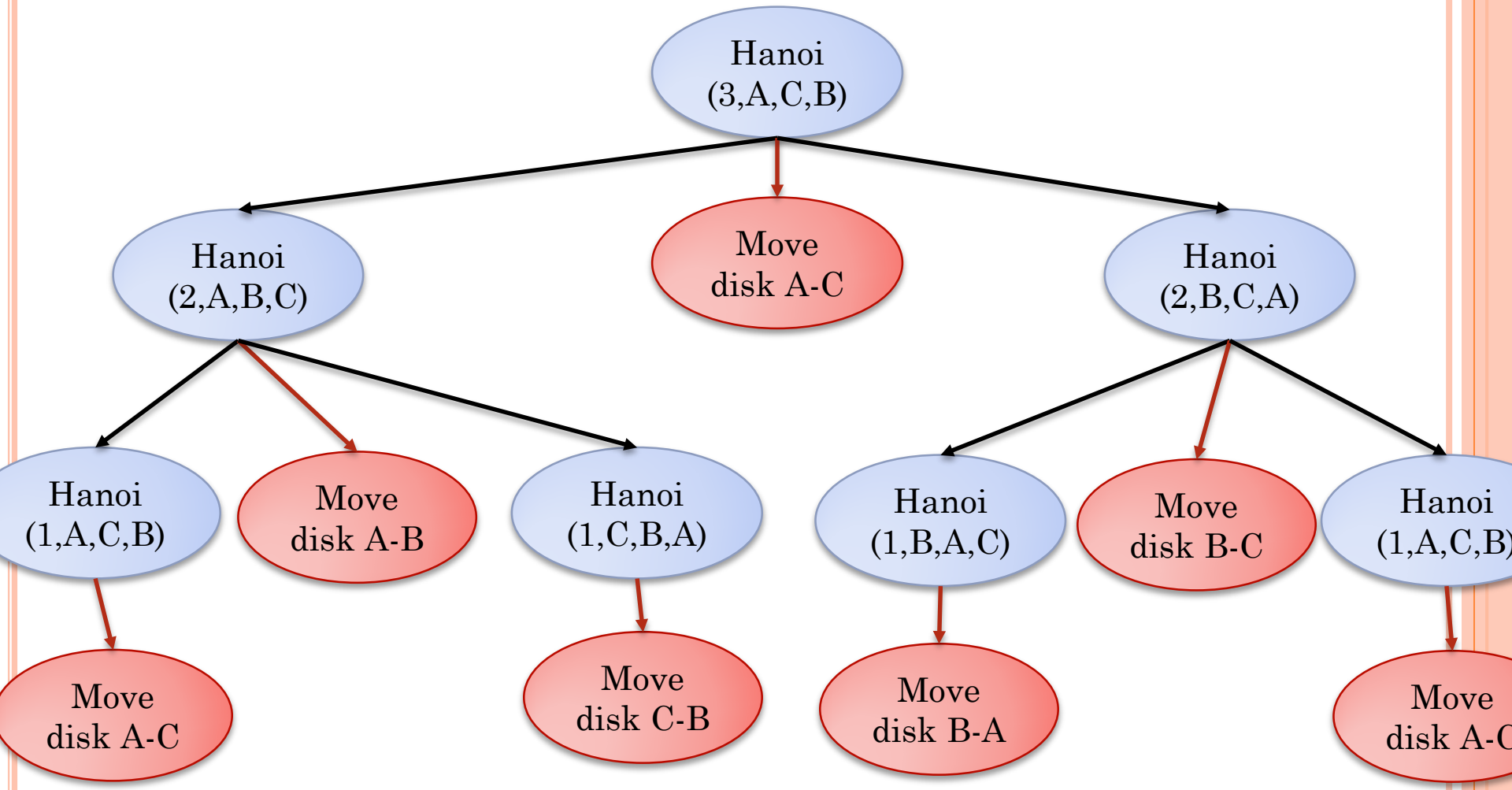


Recursion - Tower of Hanoi

```
Step 1 – Move n-1 disks from source to aux
Step 2 – Move nth disk from source to dest
Step 3 – Move n-1 disks from aux to dest

START
Procedure Hanoi(disk, source, dest, aux)

   IF disk == 1, THEN
      move disk from source to dest
   ELSE
      Hanoi(disk - 1, source, aux, dest)     // Step 1
      move disk from source to dest          // Step 2
      Hanoi(disk - 1, aux, dest, source)     // Step 3
   END IF

END Procedure
STOP
```


Reference: https://www.tutorialspoint.com/data_structures_algorithms/tower_of_hanoi.htm

TRACE OF RECURSION FOR 3 DISKS



Complexity Analysis Exercises

Consider the following code,

```
int recursiveFun1(int n)
{
    if (n <= 0)
        return 1;
    else
        return 1 + recursiveFun1(n-1);
```

What's the big O complexity ?

Recursion Exercises

Question 1:

The Greatest Common Divisor (GCD) of two integers, when at least one of them is not zero, is the largest positive integer that divides the two numbers without a remainder. Given integers m and n, find the GCD of m and n, recursively.

Recursion Exercises

Question 2:

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. Write a recursive algorithm to determine whether integer number *n* is a prime.

Proof of Big-Oh complexity

Question 3:

Prove that the running time

$T(n) = n^3 + 20n + 1$ *is* $O(n^4)$

Proof of Big-Oh complexity

Question 4:

Prove that the running time

$T(n) = n^3 + 20n + 1$ *is* $\Omega(n^2)$

Big-Oh complexity calculation

Question 5:

Show that Big O is transitive by relation.

That is, if f(n) = O(g(n)) and g(n) = O(h(n)), then, f(n) ≤ O(h(n))