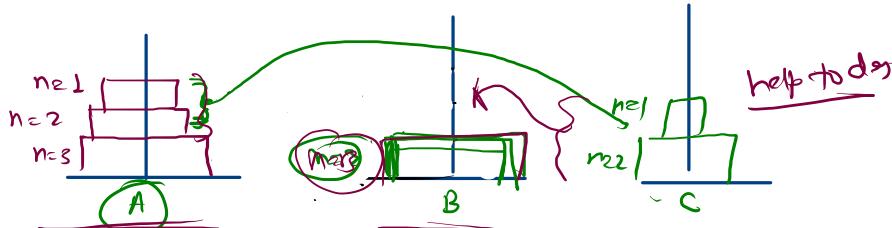


Tower of Hanoi -

$n=3$
no of disc
Set of Rules ↗

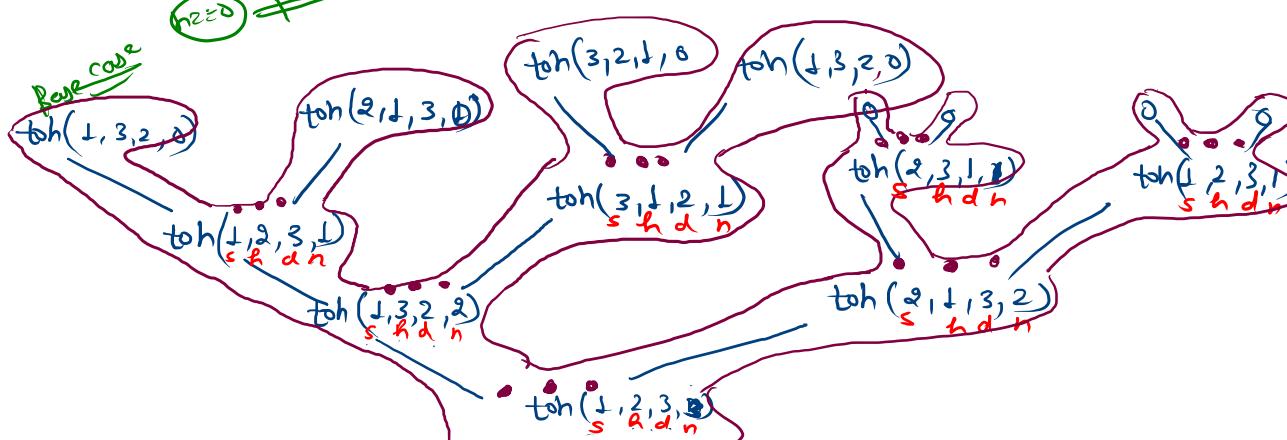


Expectation → toh(3) → It will print steps to move 'n' disks from A to B using C with follow the set of rules.

faith → toh(2) → It will definitely print the steps to move (n-1) disks from A to C } n-1 disk
print by myself → move nth disk from A to B Done by myself
toh(2) → move from C to B & print few steps

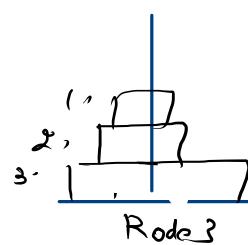
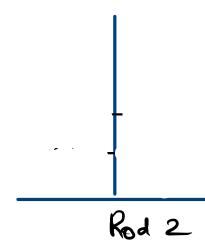
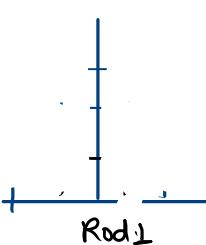
```
public static void toh(String src, String hlp, String des, int n) {
    //Write your code here.
    toh(src, des, hlp, n - 1);
    System.out.println("Move disk " + n + " from rod " + src + " to rod " + des);
    toh(hlp, src, des, n - 1);
```

main toh(1, 2, 3, n) i.e. dim is present



Move disk Gap2 from rod Gap0 to rod Gap3

Gap1 Gap2 Gap3
1, 1 → 3
2, 1 → 2
1, 3 → 2
3, 1 → 3
1, 2 → 4
2, 2 → 3
1, 1 → 3



`data = any number` → $\text{arr} = \{10, 20, 30, 40, 50, 60, 70, 80\}$

Task → If data present in array → return true
otherwise return false → present [not present]

Ques

→ Recursion ↗
 $\left. \begin{array}{l} \text{data} = 30 \\ \text{data} = 35 \\ \text{data} = 49 \\ \text{data} = 60 \end{array} \right\} \begin{array}{l} \rightarrow \text{present} \\ \rightarrow \text{not present} \\ \rightarrow \text{Not present} \\ \rightarrow \text{present} \end{array}$

$\text{arr} = \{10, 20, 30, 40, 50, 60, 70, 80, 90\}$

`data = 50`

Expectation → $\text{find}(\text{arr}, 0, 50)$ → So is present or not

Faith

→ $\text{find}(\text{arr}, 1, 50)$ → It will give me result about its presence.

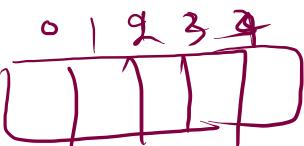
clarity ↓

Merging → Faith & Expectation

10
check yourself

$\text{find}(\text{arr}, 0, 50) \rightarrow$
check yourself that you are equal to data or not
if NOT → $\text{find}(\text{arr}, 1, 50)$

boilerplate code → $\text{recres} = \text{find}(\text{arr}, 1, 50)$
 $\text{if}(\text{recres} == \text{true}) \{$
return true;
 $\text{else} \{$
check yourself



index
j=3
return false

↓
↓
↓
↓



```
public static boolean find1(int[] arr, int idx, int data) {
    if (idx == arr.length) {
        return false;
    }
    // check yourself
    if (arr[idx] == data) {
        return true;
    }
    return find1(arr, idx + 1, data);
}
```

Base

Pre Area

Bottom up Approach

```
public static boolean find2(int[] arr, int idx, int data) {
    if (idx == arr.length) {
        return false;
    }
    boolean recRes = find2(arr, idx + 1, data);
    if (recRes == true) {
        return recRes;
    }
    return arr[idx] == data;
}
```

Base

Post Area

Top down Approach



$\text{gndx} \rightarrow$
 $\text{arr} = \{ 10, 20, 30, 40, 30, 30, 50, 60, 30, 70, 30, 30, 80 \}$

$\text{data} = 30$

$\text{ans} \rightarrow \underline{\text{Result}}$

$\underline{\text{arr}}[\underline{\text{indx}}] = \{ 2, 4, 5, 8, 10, 11 \}$

Signature:

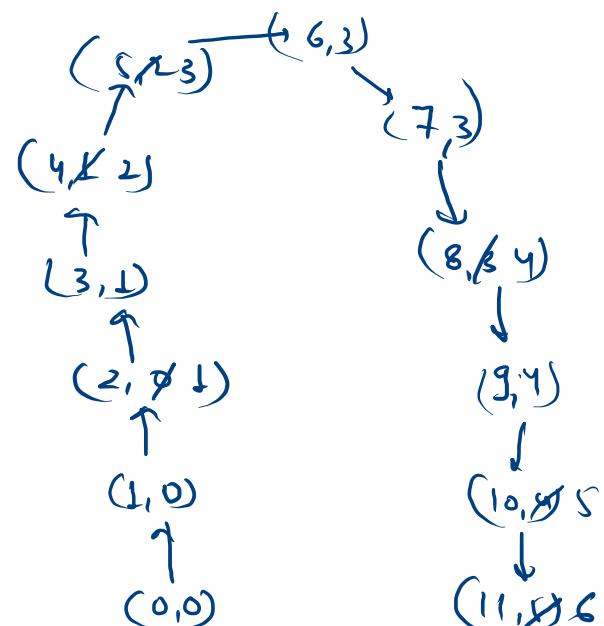
$\text{int}[] \text{ arr}(\underline{\text{arr}}, \underline{\text{data}}, \underline{\text{indx}}, \underline{\text{countsofar}})$

Recursion → Munka \leftarrow Bottom Up

Munka \leftarrow Top Down

arr
 $\text{data} = 30$

$\text{countssofar} \rightarrow$ No of occurrences of $\underline{\text{data}}$



Bug \leftarrow Car
 $\text{index} = \text{arr.length}$
~~return~~
base case
 $\text{arr}[\text{index}] = \text{data}$
array size (csf)
Count so far

$\text{arr} = \{ 10, 20, 20, 20, 30, 40, 30 \}$

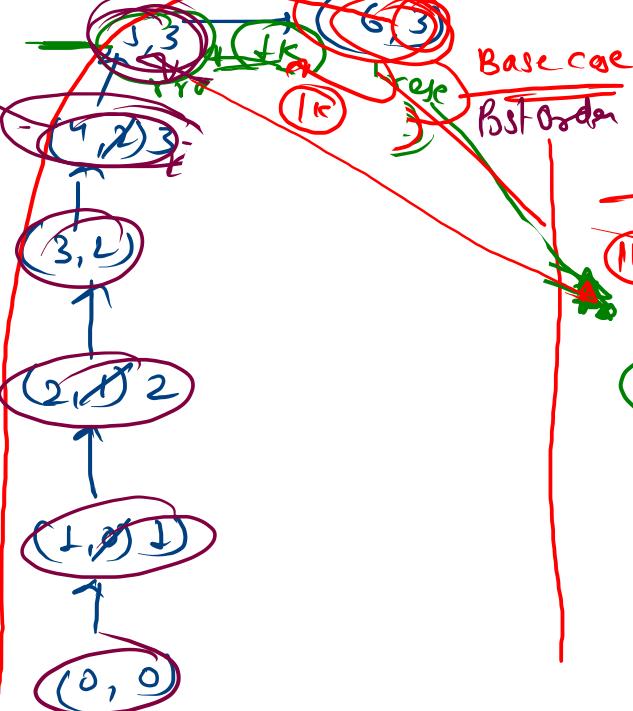
$(\underline{\text{arr}}, \underline{\text{data}} | \underline{\text{indx}}, \underline{\text{csf}})$

$\text{data} = 20$

Global Set

How

count so far manage



Heap memory

