Recursion

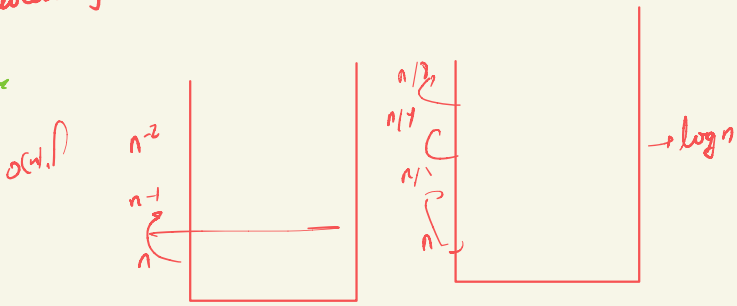
1) expectation → whatever is asked in the question

2) Faith → my faith in the function

3) do some work to reach your expectation from your faith.

4) try smaller and find base case.

⇒ print numbers in increasing order.
⇒ " " " decreasing order.
⇒ factorial of a number

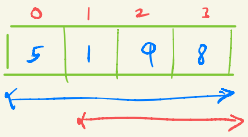


Recursion in Arrays

Faith

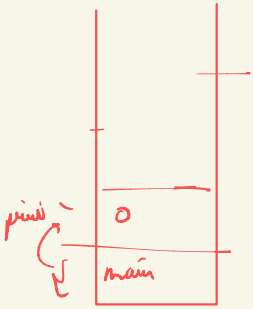
```
print(arr, idx+1);
```

8
9
1
5

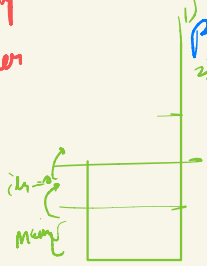



```
print(arr, idx)

if(idx == arr.length)
    return;
recursion on the way down
→ pre order
print(arr, idx+1);
Syscout(arr[idx]);
```

5
1
9
8



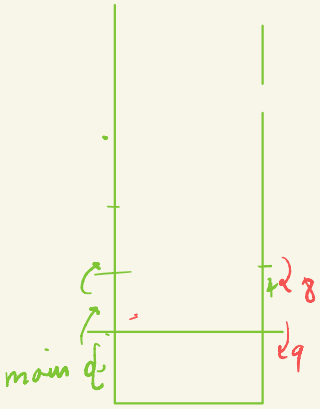
→ recursion on the way up
→ preorder



8
9
1
5

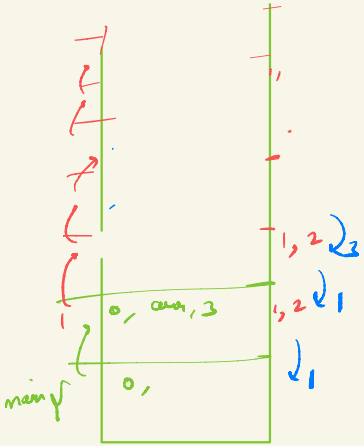
→ recursion in the way down

```
public static int maxOfArray(int idx, int[] arr){
1   if(idx==arr.length){
        return 0;
    }

2   int sAns = maxOfArray(idx+1, arr);
3   int ans=Math.max(arr[idx],sAns);

    return ans;
};
```



0	1	2	3
9	0	8	7

0	1	2	3	4
2,	3,	0,	3,	4

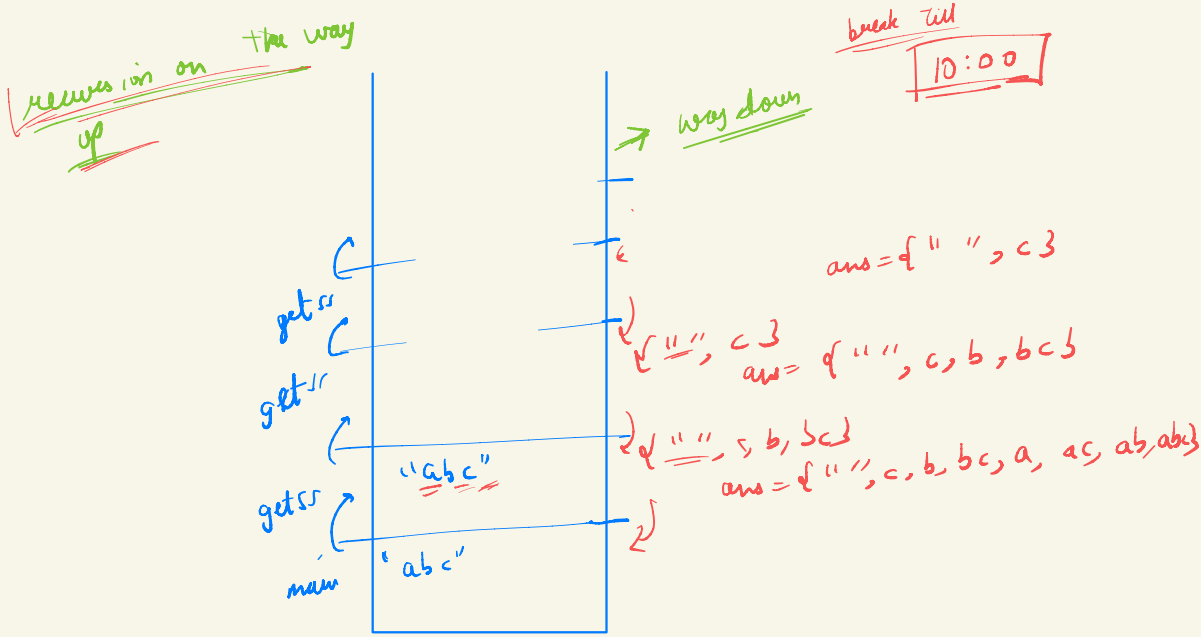


recursion in the way down →

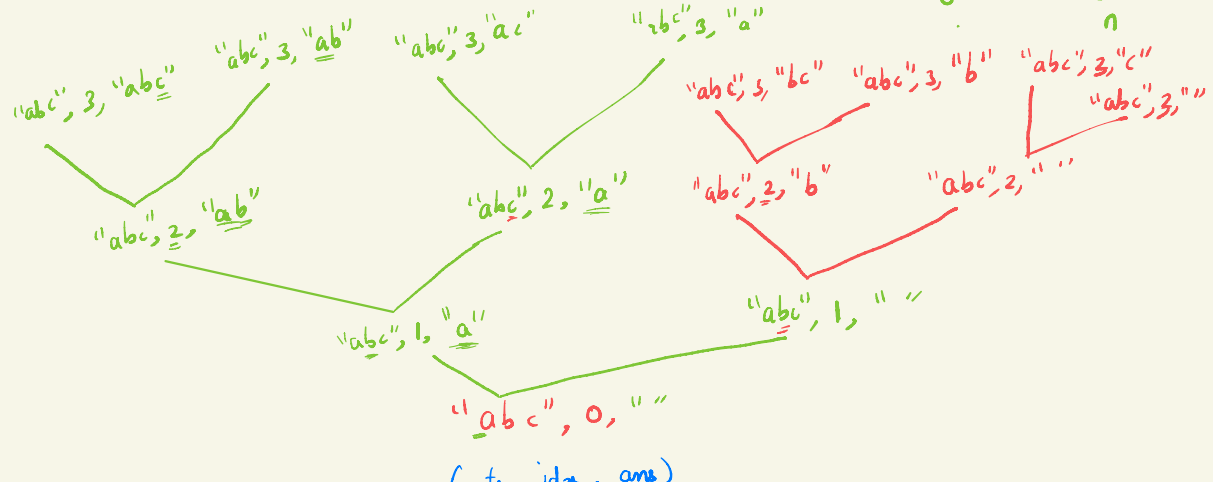
```
public static int firstIndex(int idx, int[] arr, int tar){
1)  if(idx == arr.length){
        return -1;
    }

2)  int sAns = firstIndex(idx+1, arr, tar);
    int ans = 0;
    if(arr[idx] == tar){
        ans = idx;
    } else {
        ans = sAns;
    }

    return ans;
}
```

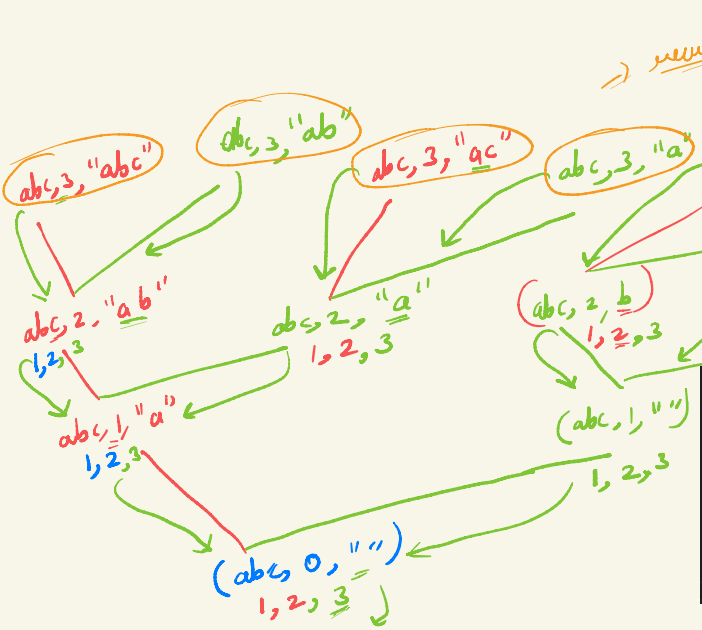
Count → 2 ∴ there is 2 occurrence before this curr. occurrence

0	1	2
		4



a	b	c
y	y	y
y	y	n
y	n	y
y	n	n






```java
public static void printSS(String str, int idx, String ans){
    if(idx==str.length()){
        System.out.println(ans);
        return;
    }

    // say yes to curr char
    printSS(str, idx + 1, ans+str.charAt(idx));

    // say no to curr char
    printSS(str, idx + 1, ans);
}
```