Array → Idx

Recursion → Array, idx

Display using Recursion

arr =

0	1	2	3	4	5	6
10	20	30	40	50	60	70

p s v display (arr, idx){

E

f(arr,0) => 10

20
30
40
50
60
70

F

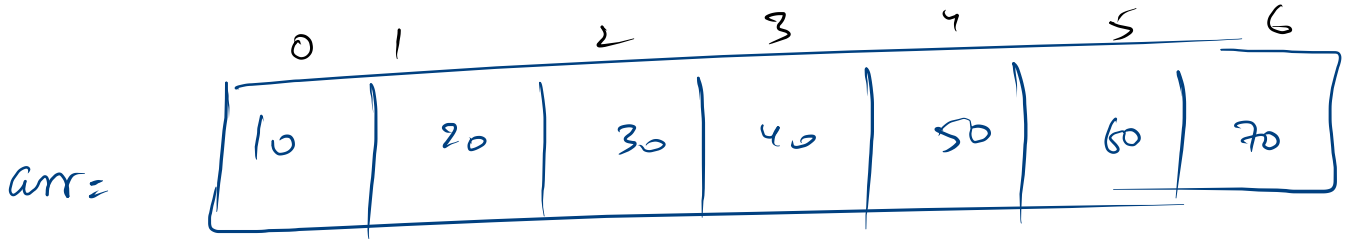
f(arr,1) => 20
30
40
50
60
70

}

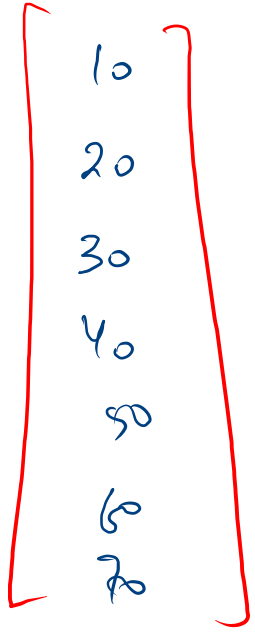
f(arr,0) => print(arr[0]) W
E
f(arr,1) ✓
F

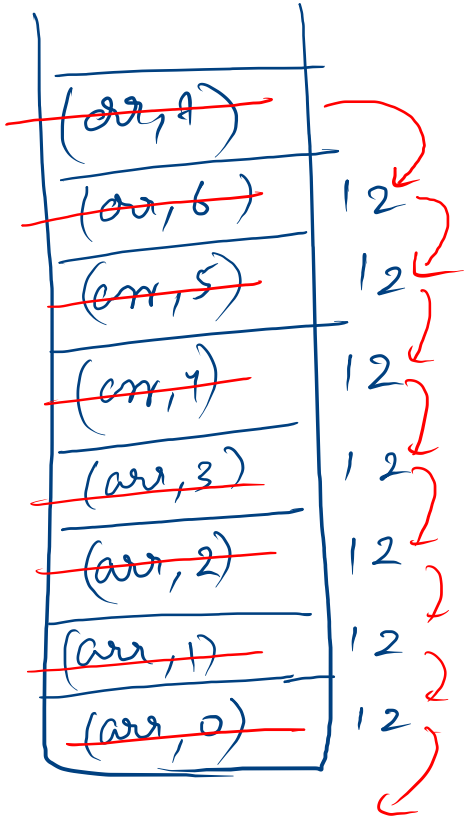
work + faith → Expectation

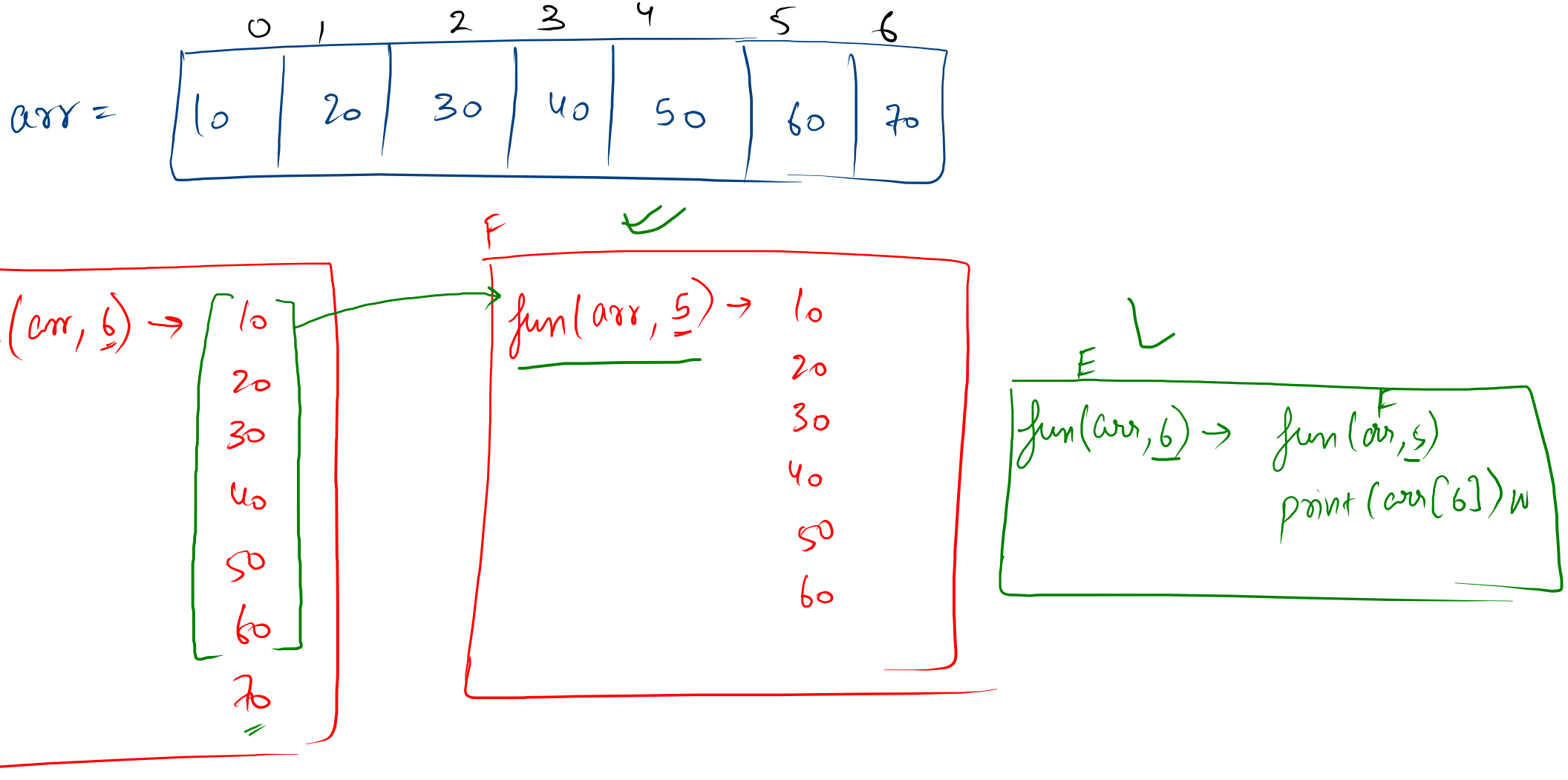
```java
public static void displayArr(int[] arr, int idx){
    System.out.println(arr[idx]); -①
    displayArr(arr,idx+1);  -②
}
```



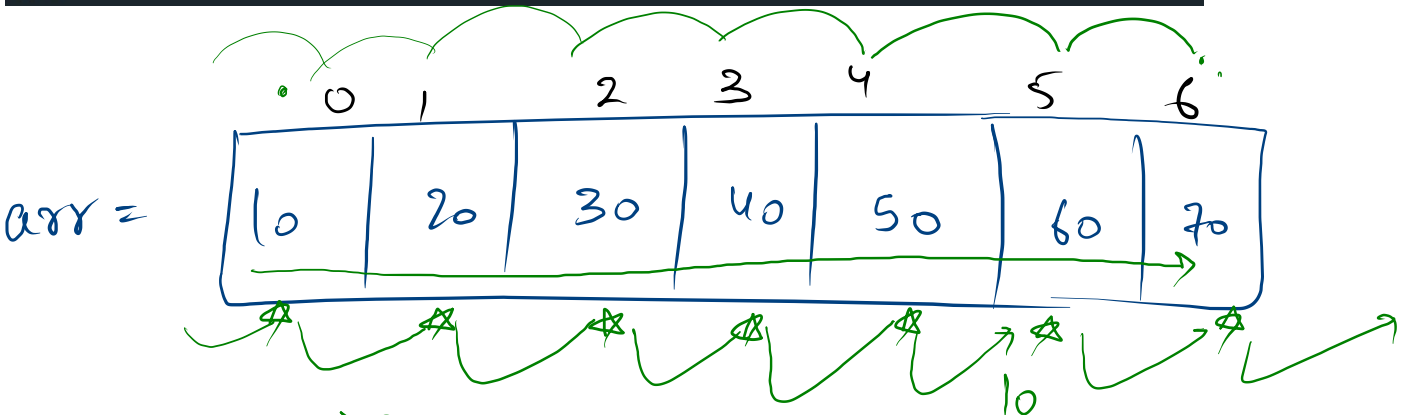

```
if(idx == arr.length){
    return;
}
```





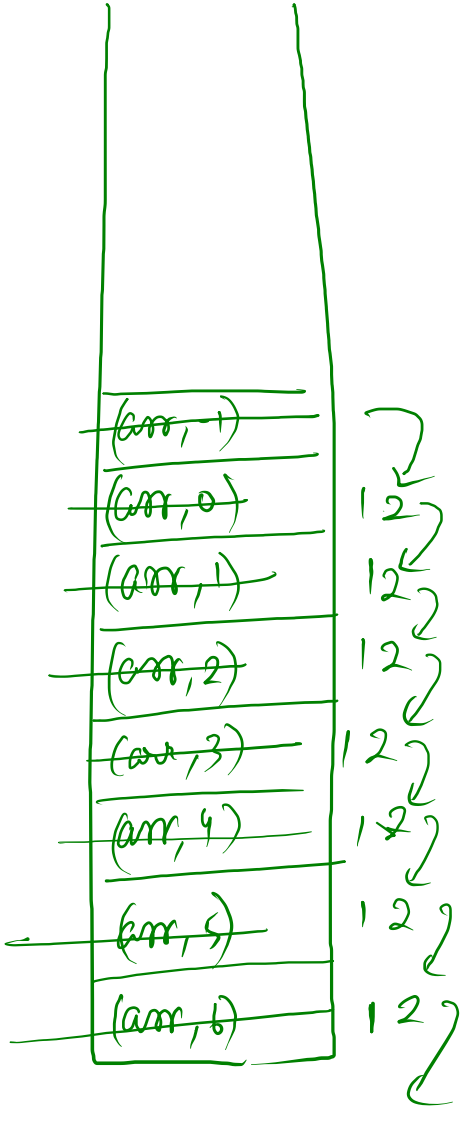

0 1 2 3 4 5 6
arr =
10 20 30 40 50 60 70
E
fun(arr, 6) →
10
20
30
40
50
60
70
F
fun(arr, 5) →
10
20
30
40
50
60
E
F
fun(arr, 6) → fun(arr, 5)
print(arr[6])

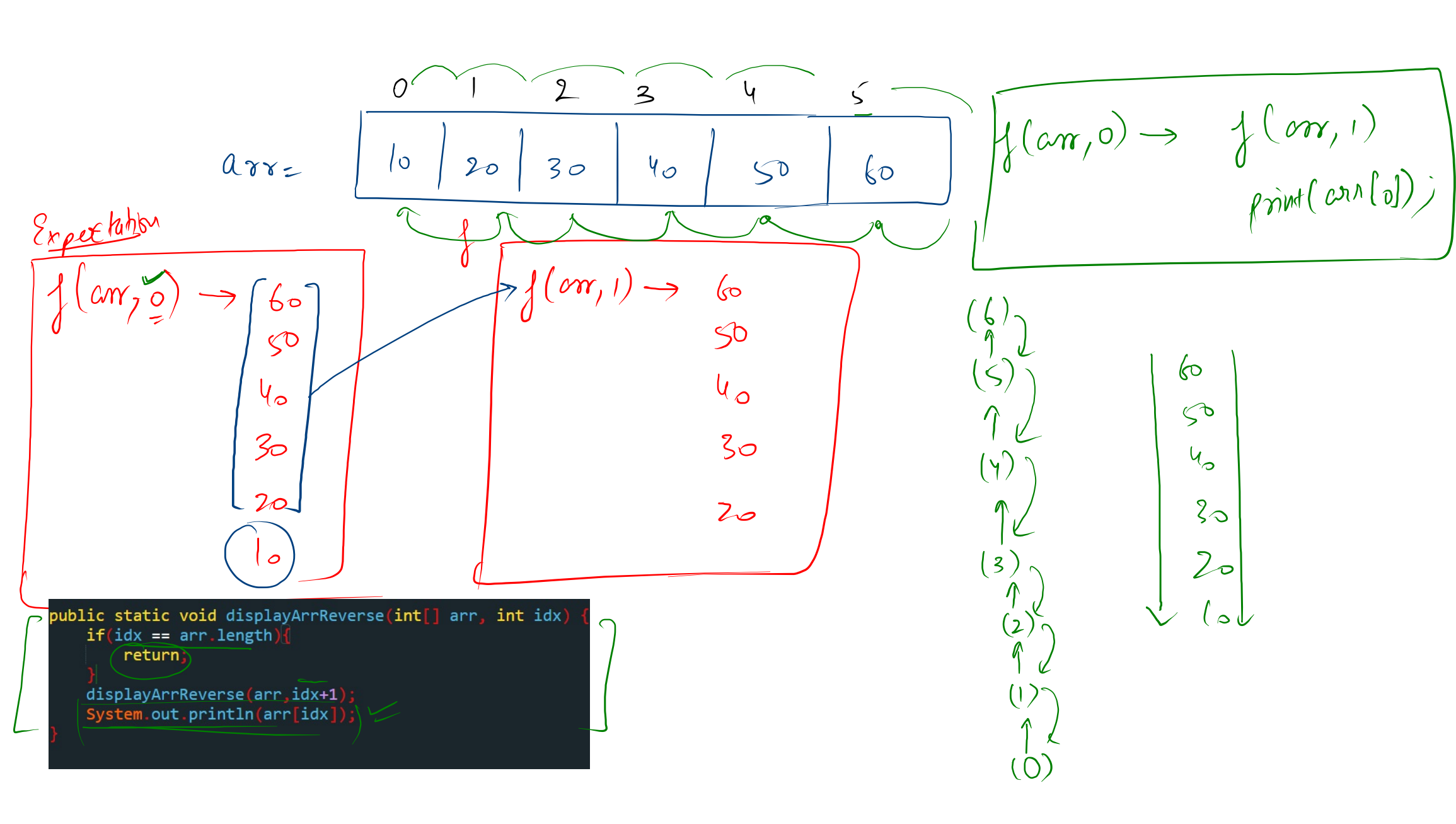
```java
public static void displayArr2(int arr[] ,int idx){
    displayArr2(arr,idx-1); // 1
    System.out.println(arr[idx]); // 2
}
```



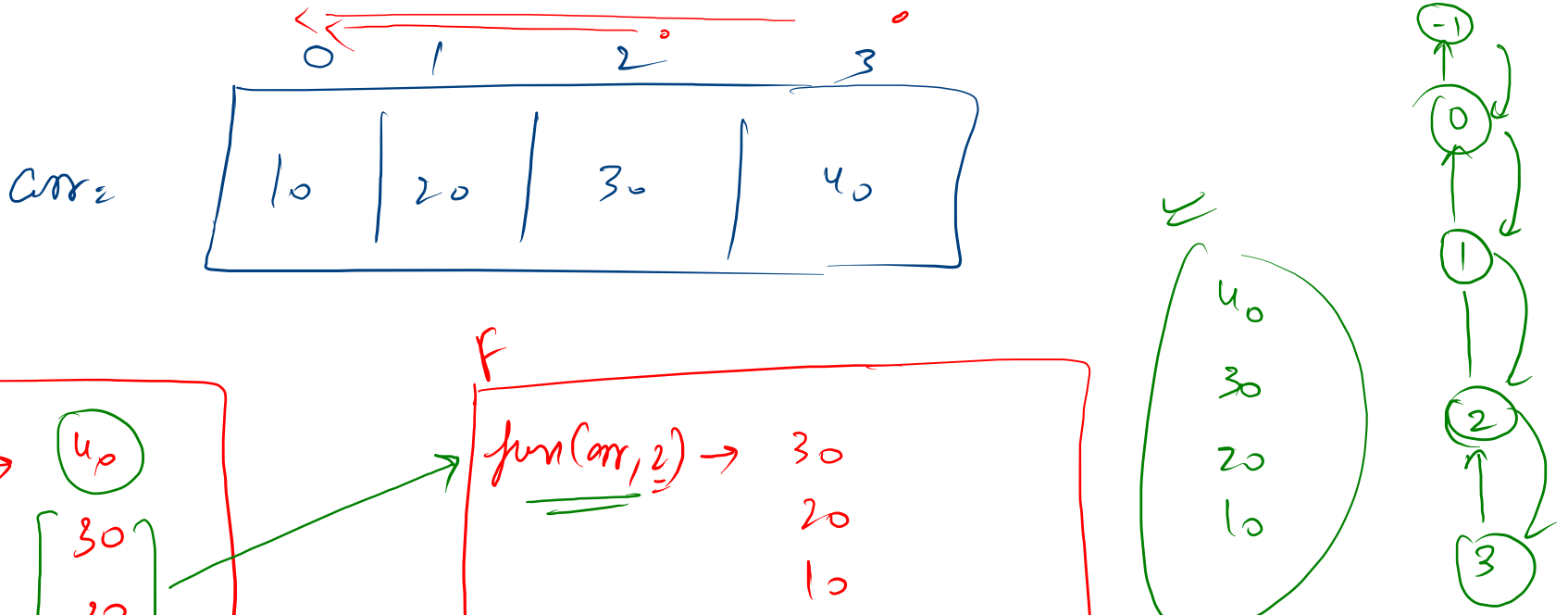

```
if( idx == -1){
    return;
}
```

10
20
30
40
50
60
70

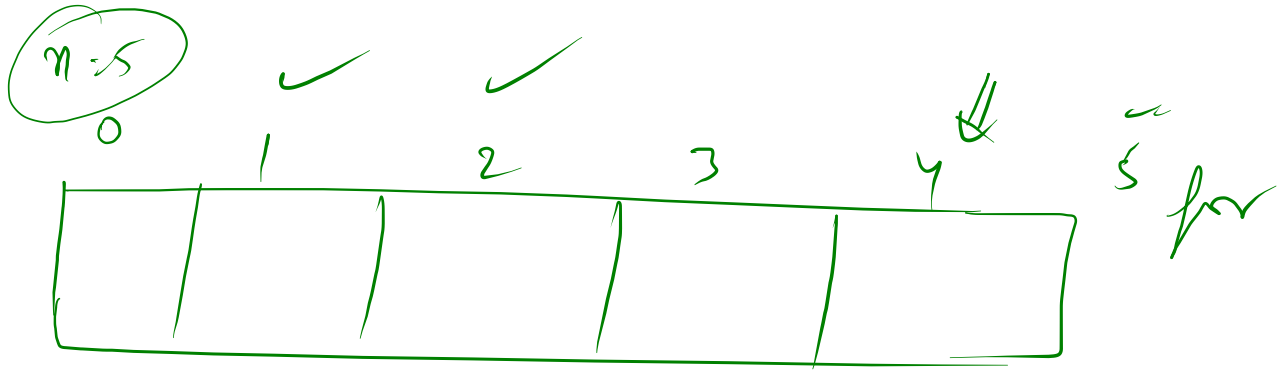


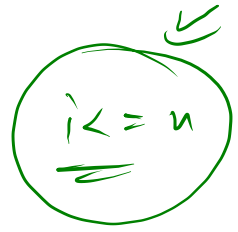



```java
public static void displayArrReverse(int[] arr, int idx) {
    if(idx == arr.length){
        return;
    }
    displayArrReverse(arr,idx+1);
    System.out.println(arr[idx]);
}
```




```java
public static void displayArrReverse1(int[] arr, int idx) {
    if(idx == -1){
        return;
    }
    System.out.println(arr[idx]);
    displayArrReverse1(arr,idx-1);
}
```



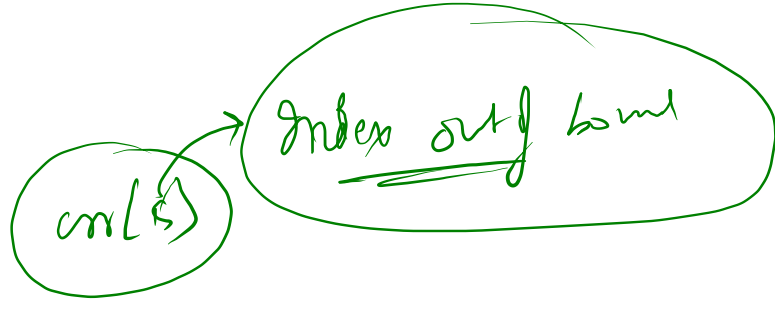


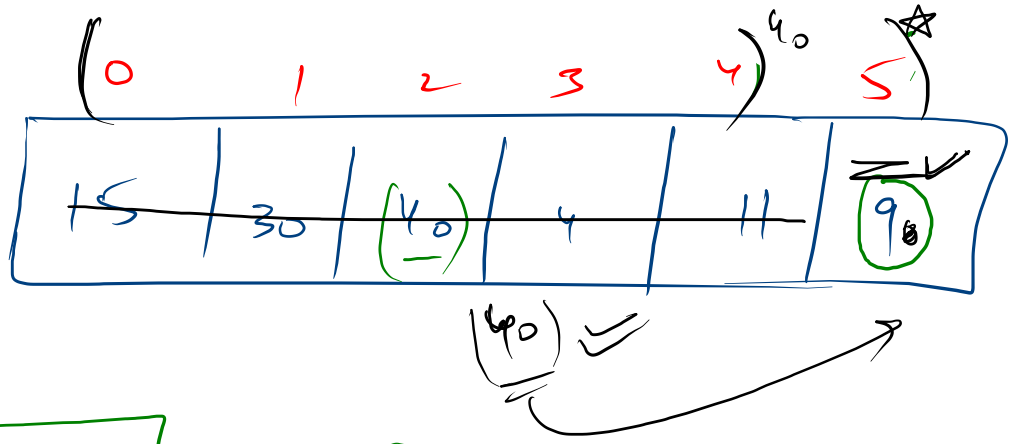

```
for(i=1 ; i<=n ; i++) {
    arr[i] =
}
```

i => 1
i => 2
i => 3

arr[1]
arr[2]
arr[3]

5





E

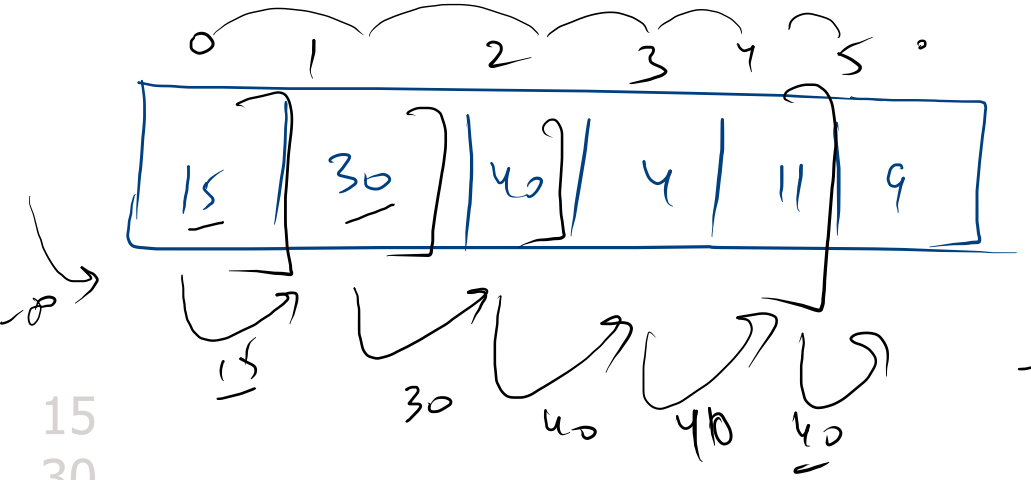
fun(arr, 5) ⇒ 40

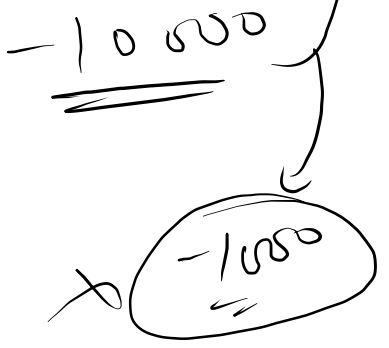
(f(arr, 4) = 40)

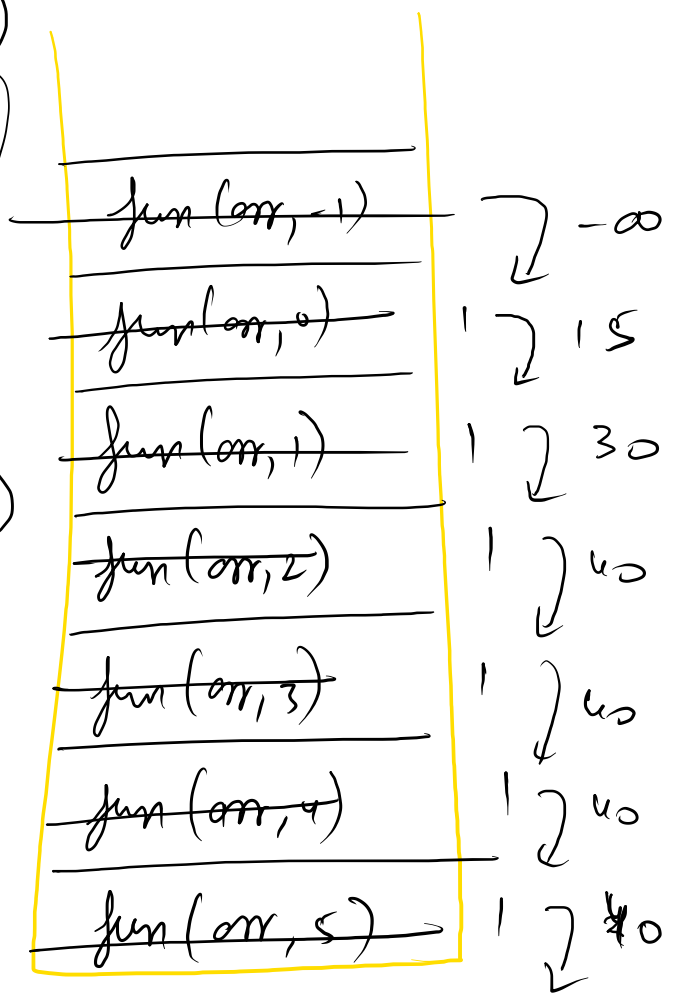
*

fun(arr, idx) ⇒ Max(arr[idx], fun(arr, idx-1))

fun(arr, 5) ⇒ Max(9, 40) ⇒ 40








```
15
30
40
4
11
9
```

```
if(idx == -1){
    return Integer.MIN_VALUE;
}
```

```
public static int maxOfArray(int[] arr, int idx){
    return Math.max(arr[idx],maxOfArray(arr,idx-1));
}
```

max of array(arr, len-1)

max of arr(arr, 0) (Do yourself)

(min of arr(arr, len-1)) (=)

(min of arr(arr, 0)) (=)

(Math.min)

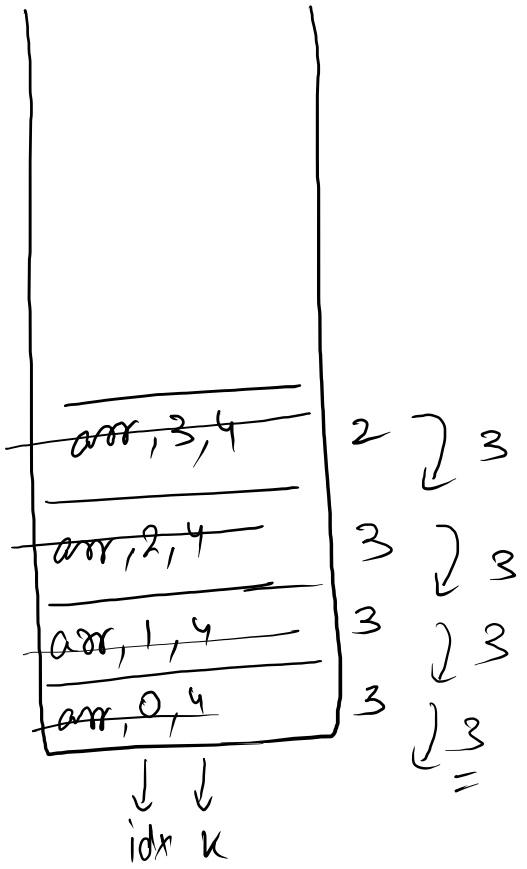
0 → n

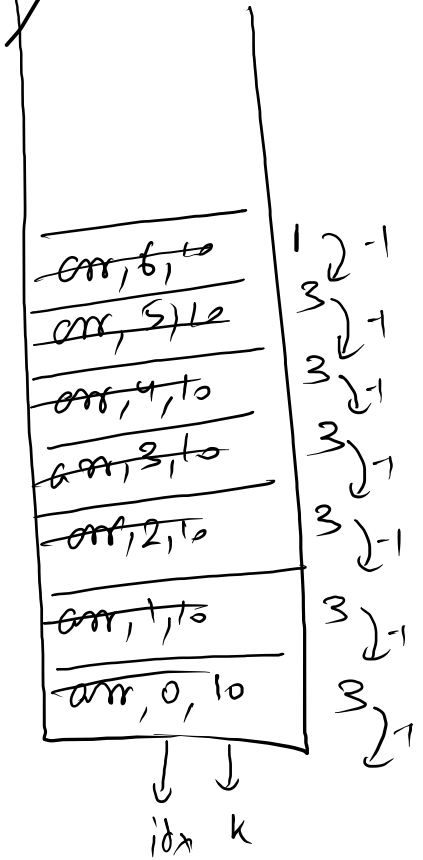
arr =

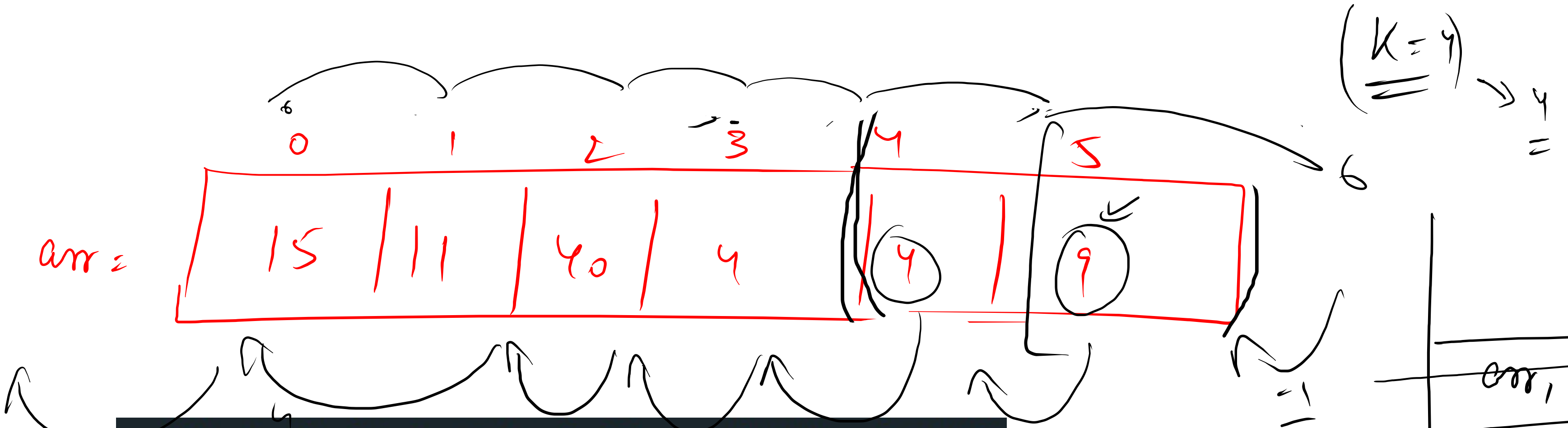
0	1	2	3	4	5
15	11	40	4	4	9

```java
public static int firstIndex(int[] arr, int idx, int k){
    if(idx == arr.length){
        return -1;          // 1
    }
    if(arr[idx] == k){
        // first occurance
        return idx;         // 2
    }

    int res = firstIndex(arr,idx+1,k);   // 3
    return res;
}
```

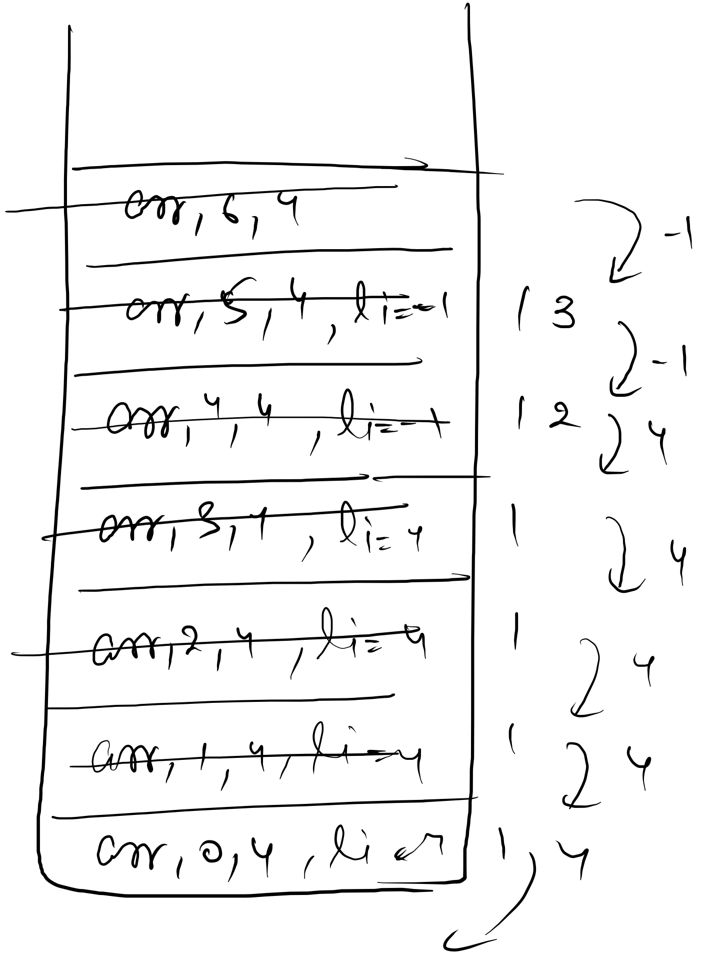






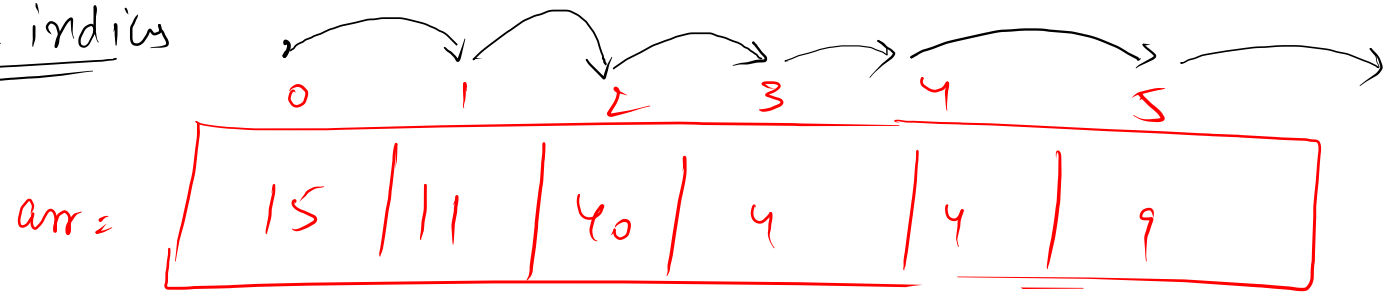

```java
public static int lastIndex(int[] arr, int idx, int k){
    if(idx == arr.length){
        return -1;
    }
    int li = lastIndex(arr,idx+1,k);

    if(li == -1){
        if(arr[idx] == k){
            return idx;
        }else{
            return -1;
        }
    }else{
        return li;
    }
}
```

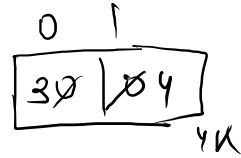


K = 4

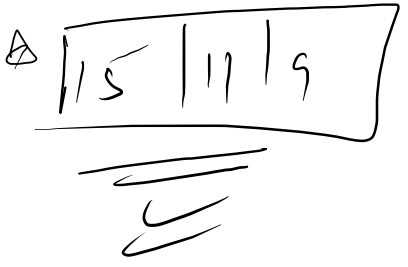
All indices




```
if (idx == a.len)
  return newint[2];
}
```



fsf => frequency so far




```
if (arr[idx] == k) {
  int res[] = allIndices(arr, k, idx + 1, fsf + 1);
  res[fsf] = idx;
  return res;
} else {
  int res[] = allIndices(arr, k, idx + 1, fsf);
  return res;
}
```