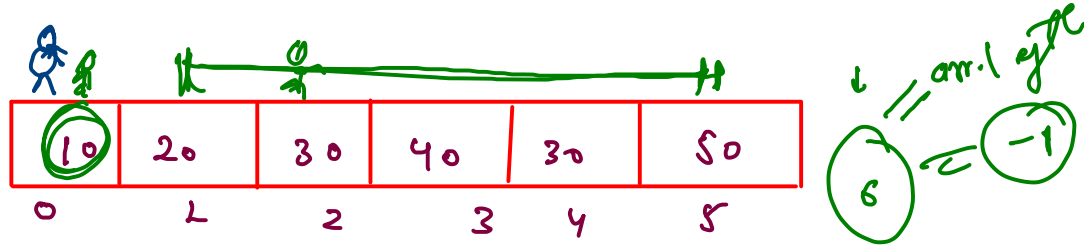


Last Index



Expectation.

$\text{last}(\text{arr}, 0)$

→

Last Index
present

→ 4

absent

→ -1

faith

→ $\text{last}(\text{arr}, 1)$

→ Last Index
absent

→ -1

Merging

→

$\text{last}(\text{arr}, 0) = li = \text{last}(\text{arr}, 1)$

if ($li == -1$) {

// check yourself.
if ($\text{arr}[li] == \text{data}$)
{
 $li = i$

return li

$li = \text{Valid Index}$

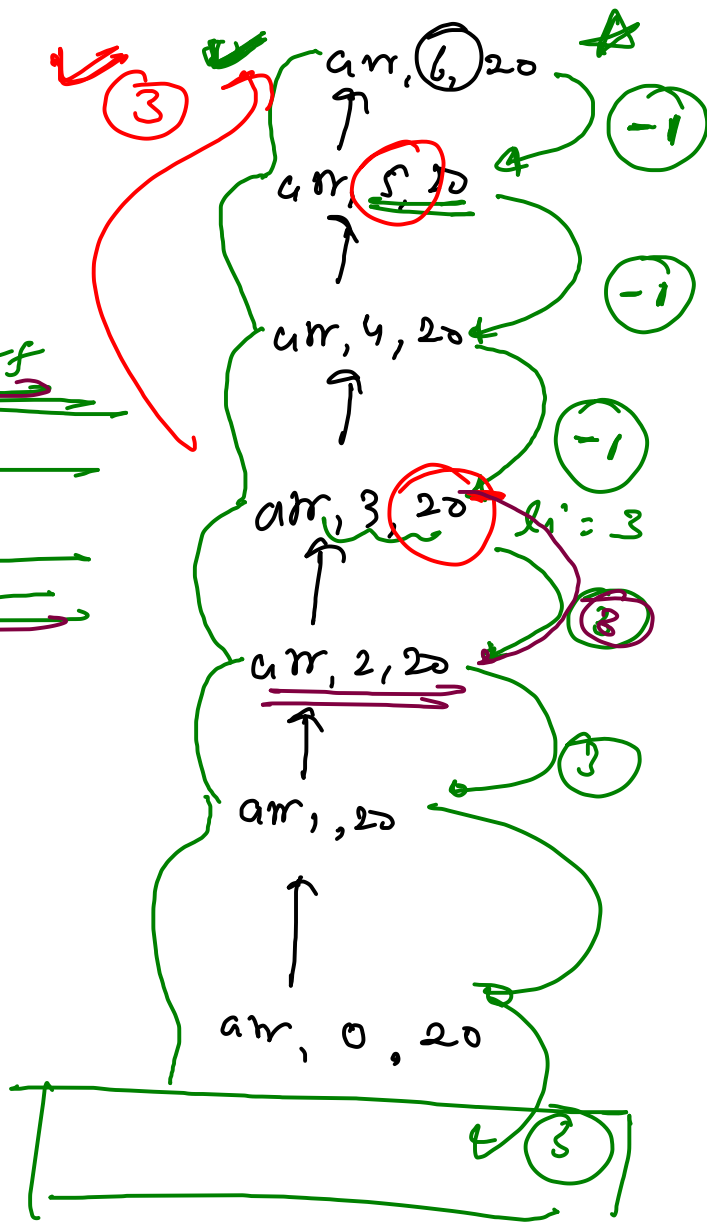
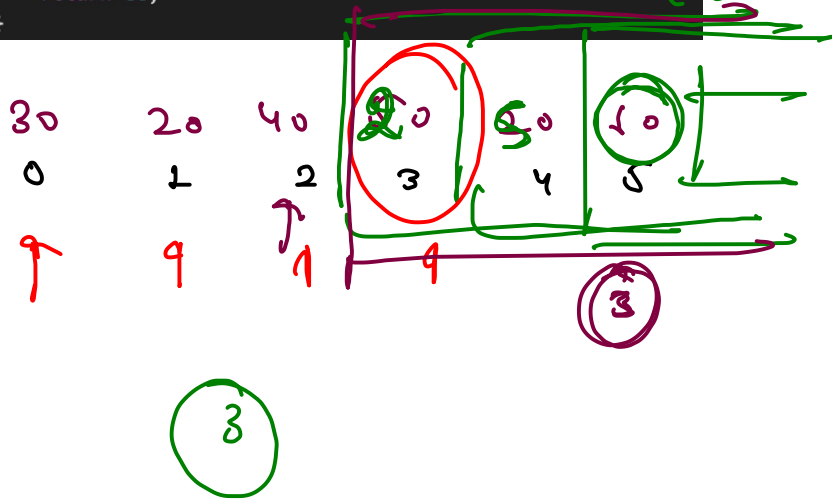
$li = -1$

```
public static int lastIdx(int[] arr, int indx, int dtf) {
    if(indx == arr.length) return -1;

    int li = lastIdx(arr, indx + 1, dtf);

    if(li == -1 && arr[indx] == dtf) {
        li = indx;
    }

    return li;
}
```



index := arr.length

Post Area

all indices \rightarrow

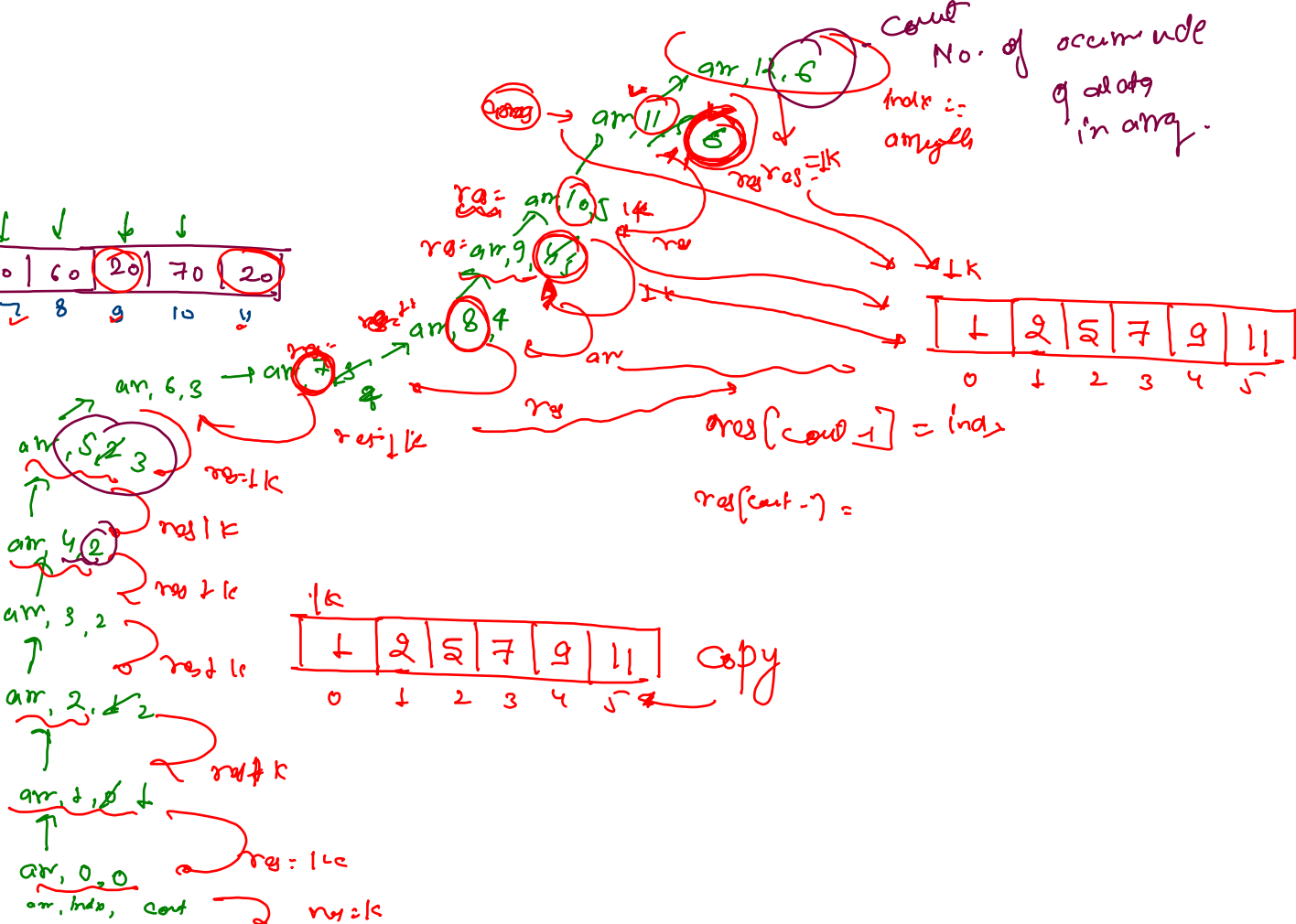
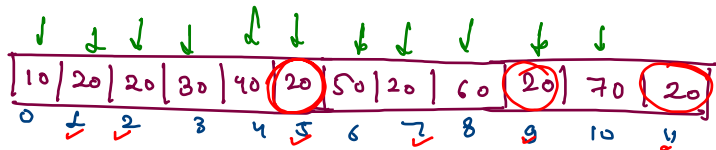
data = 20

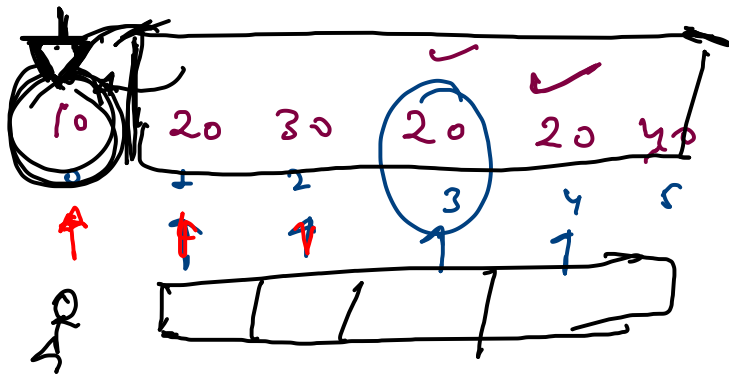
1	2	5	7	9	11
0	1	2	3	4	5

date : 20

Pre \rightarrow count increment
Post \rightarrow result making.

main =



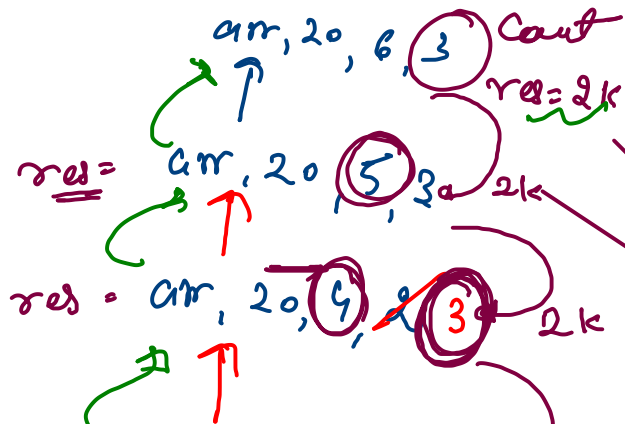


```

public static int[] allIndices(int[] arr, int data, int indx,
    if(indx == arr.length) {
        int[] bres = new int[count];
        return bres;
    }
    if(arr[indx] == data) {
        count++;
    }
    int[] res = allIndices(arr, data, indx + 1, count);
    if(arr[indx] == data) {
        res[count - 1] = indx;
    }
    return res;
}

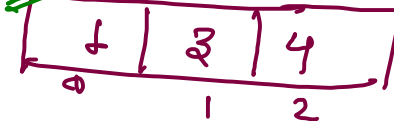
```

Expected → allIndices 1



arr, 20, 6, 3 Count
res = arr, 20, 5, 3
res = arr, 20, 4, 3
res = arr, 20, 3, 2
res = arr, 20, 2, 1
res = arr, 20, 1, 0
res = arr, 20, 0, 0

Index



fairly.

all Indices (arr, 1, 20, 1)

main

res

Expectation.

res = allIndices(arr, 0, 0, 30) →

0 - length - 1
count

[0, 1, 7, 9] =

pre

count →

faith →

allIndices(arr, count, 30) →

1 to length - 1

post

count →

0 → p

res = [0, 1, 7, 9]
↑

vector<int> arr;

}