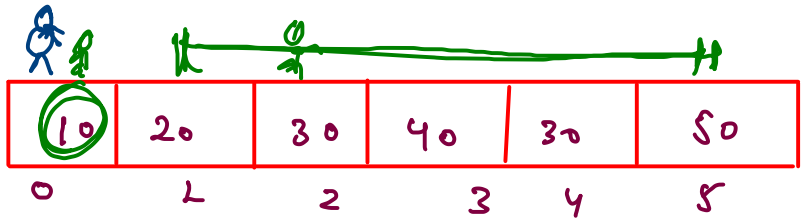
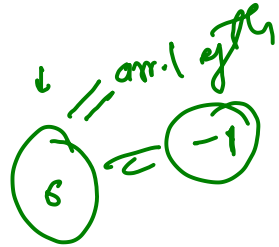
Last Index





Expectation. last(arr, 0) → last Index present → 4

absent → -1

faith. → last(arr, 1) → last Index absent → -1

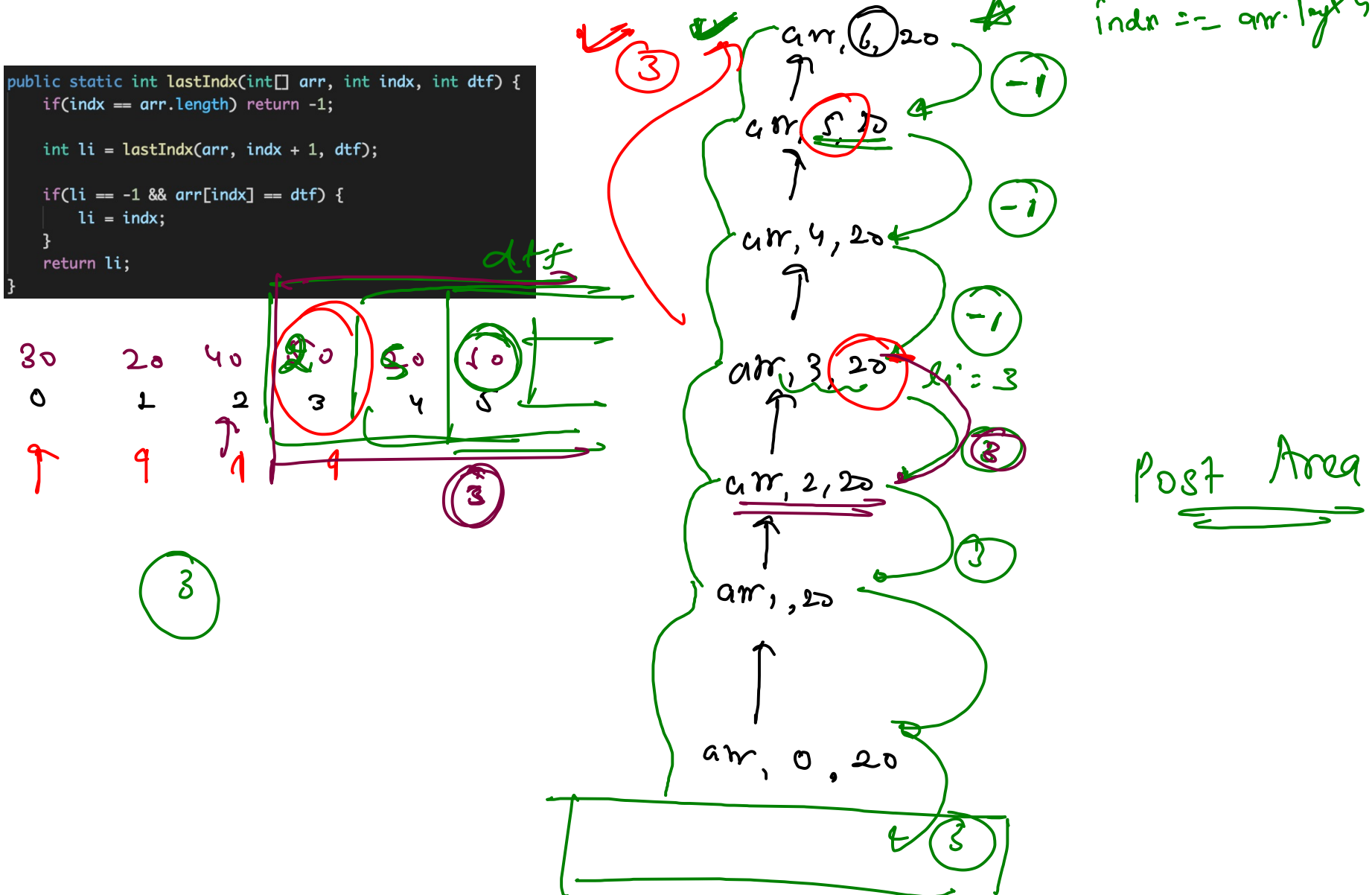
Merging → last(arr, 0) = li = last(arr, 1)

```
if(li == -1){
    // check yourself.
    if(arr[i] == data)
        li = i;
}
return li
```

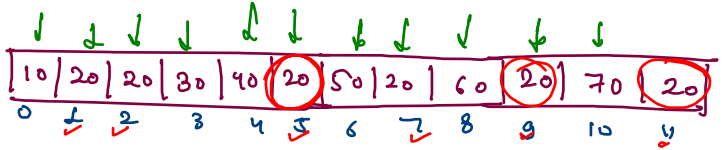
li = valid Index

li = -1

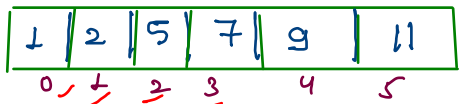
```java
public static int lastIndx(int[] arr, int indx, int dtf) {
    if(indx == arr.length) return -1;

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

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



all Indices →



data: 20

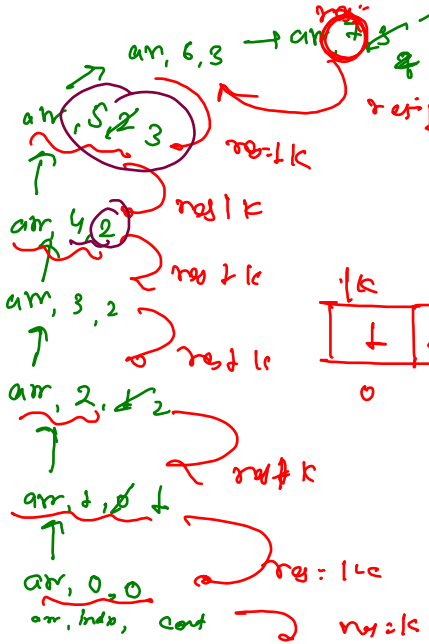


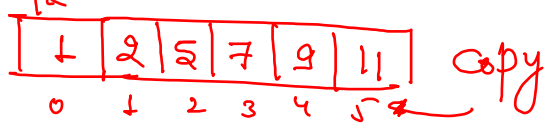
data: 20

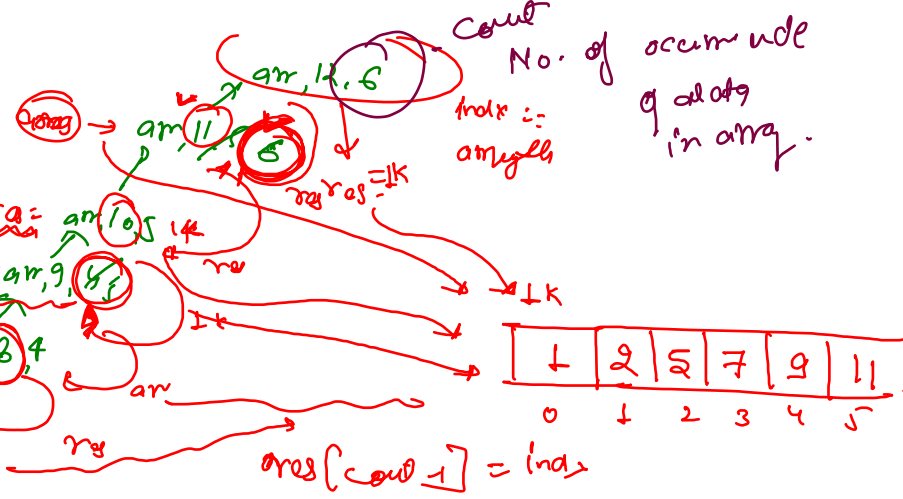
Pre → count Ignore

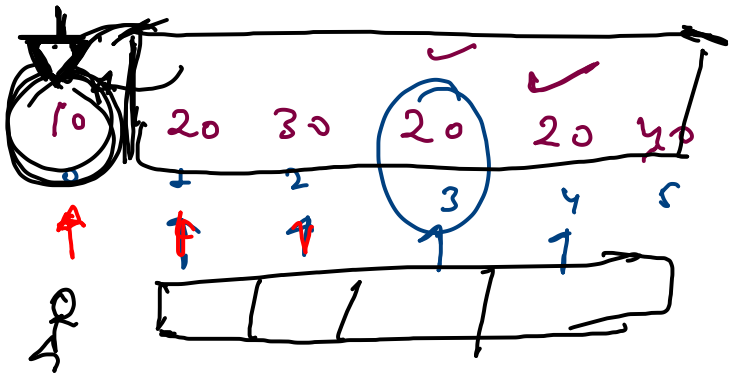
Post → Result Making.

Main =





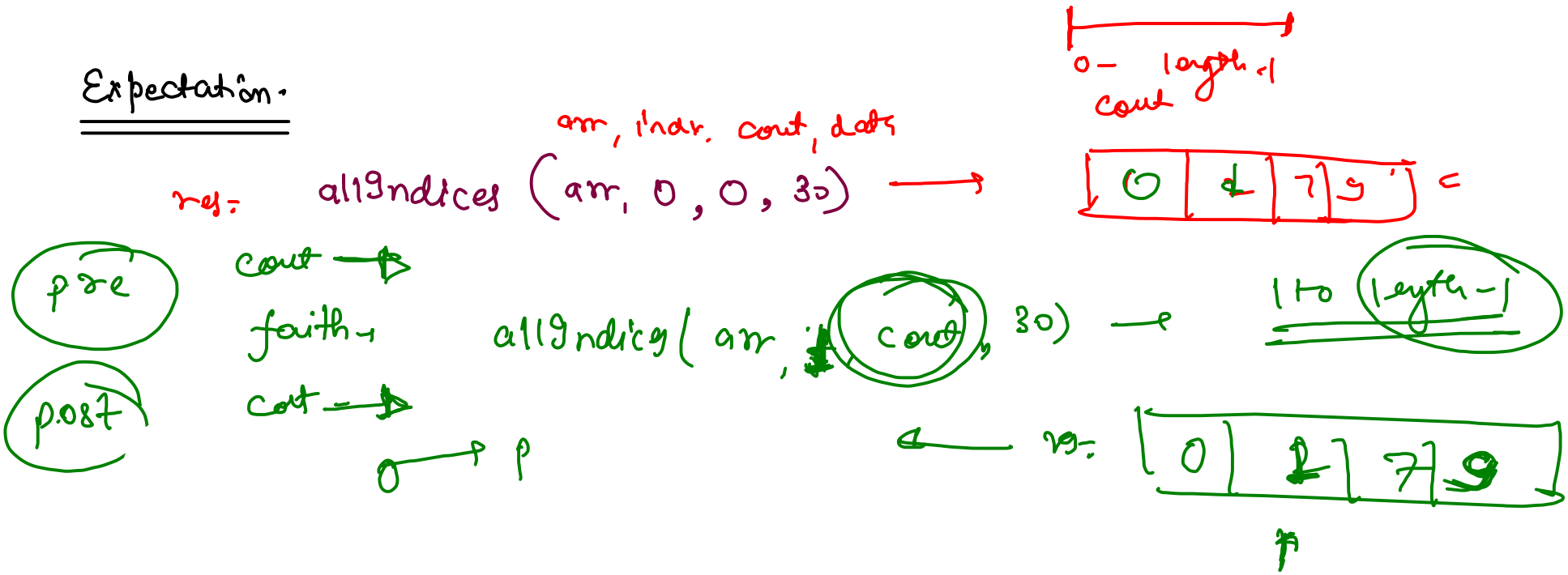





```java
public static int[] allIndices(int[] arr, int data, int indx, int count) {
    if(indx == arr.length) {
        int[] bres = new int[count];
        return bres;
    }

    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)

Expectation:



Vector<int> arr. { }