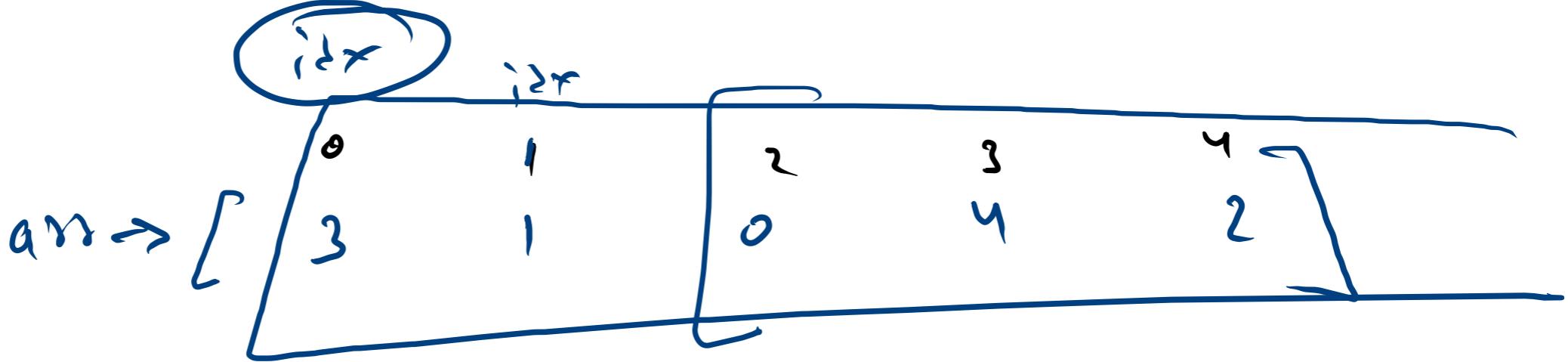


$n \rightarrow$

5



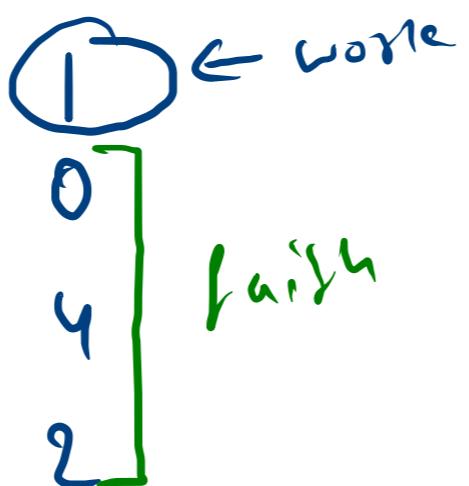
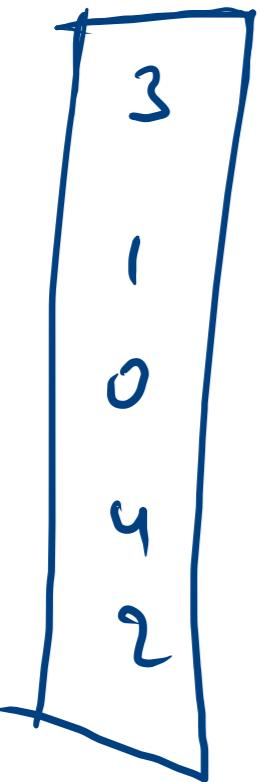
expectation

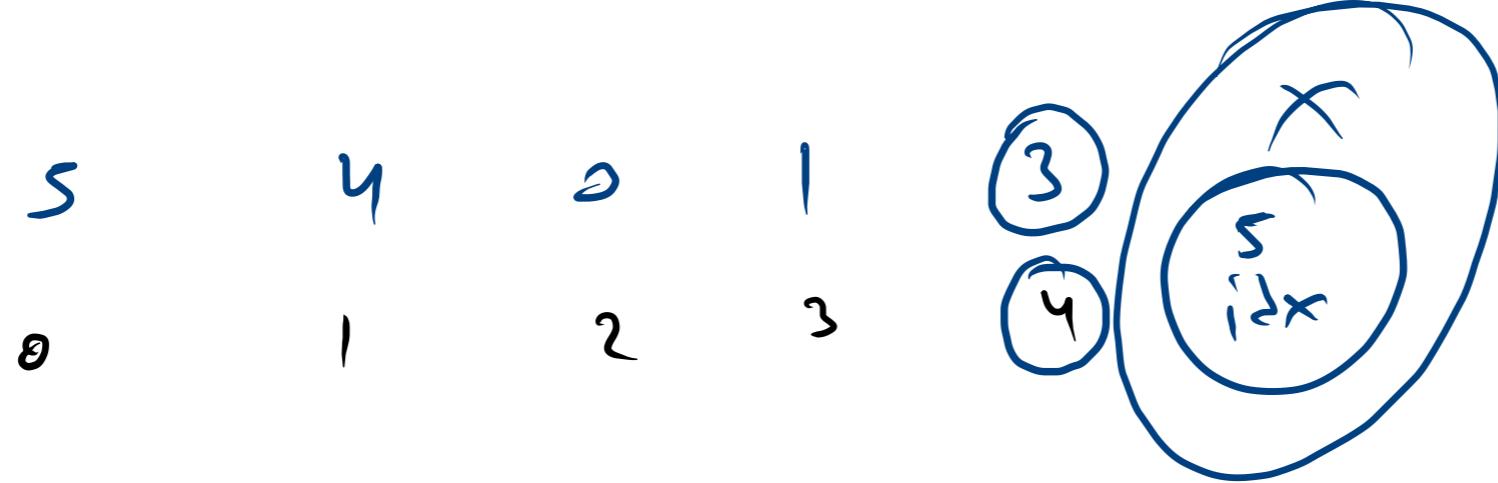
$idx \rightarrow 1$

faish

$idx \rightarrow 2$

output \rightarrow



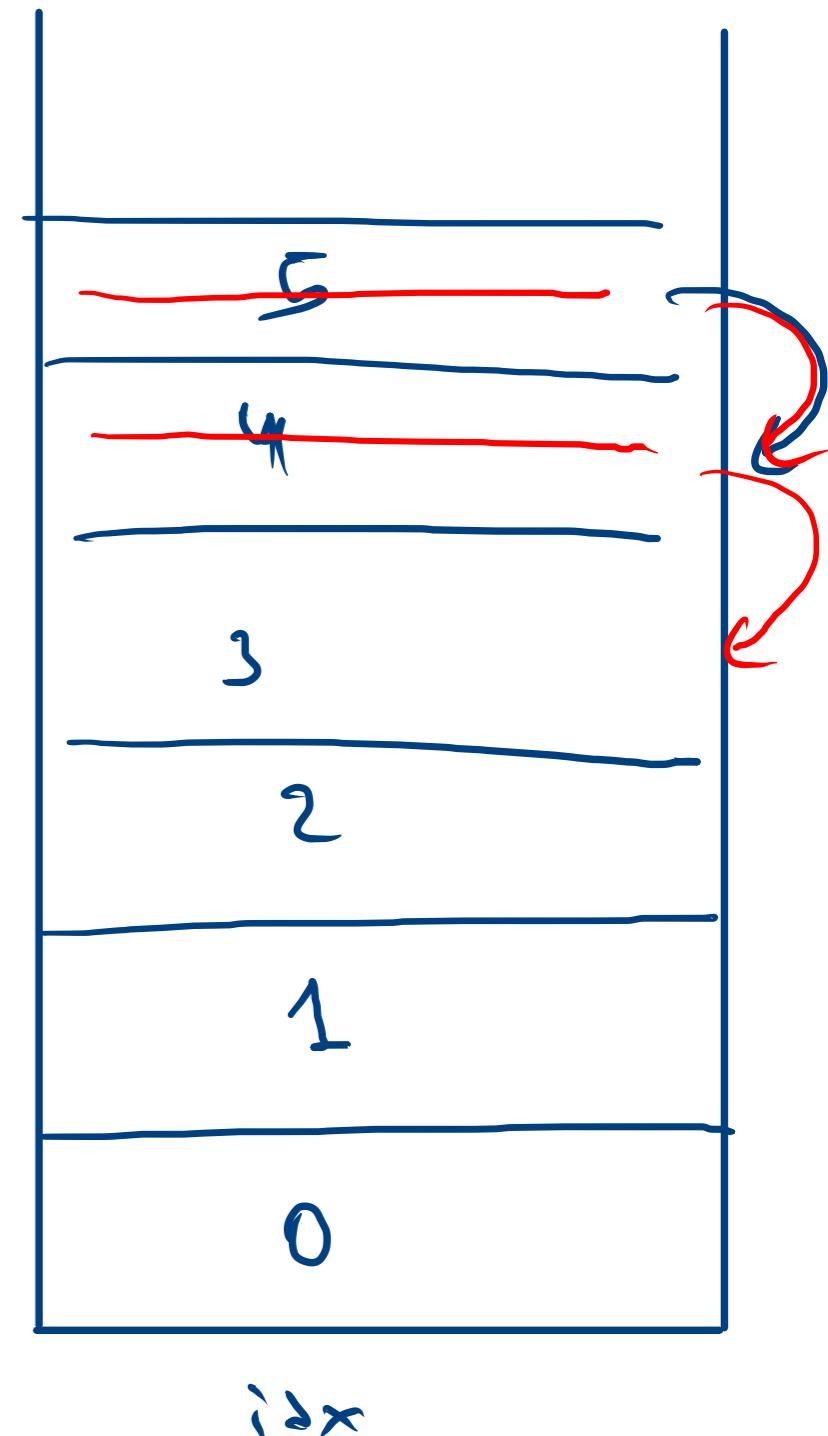


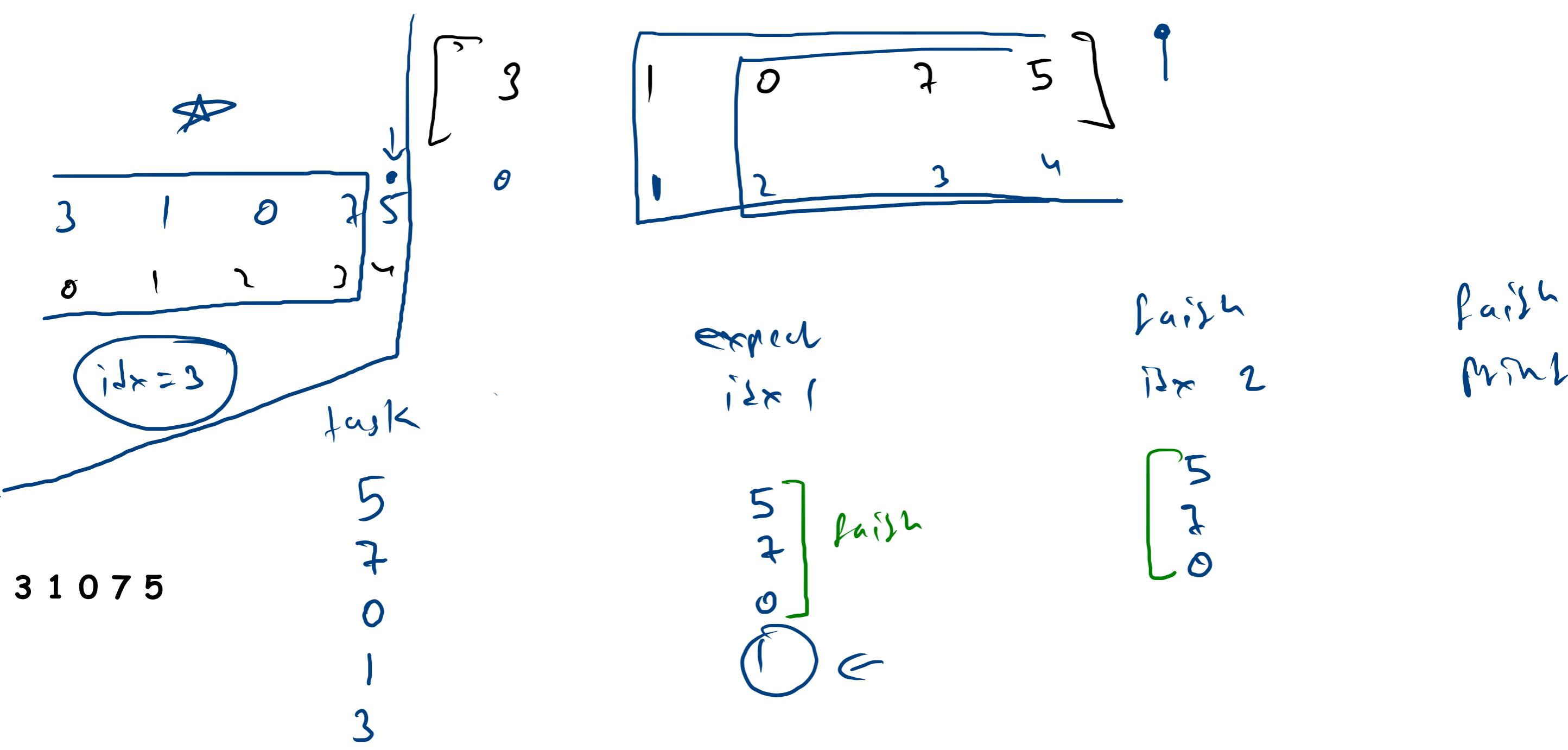
```
public static void displayArr(int[] arr, int idx){
    System.out.println(arr[idx]); ←
    displayArr(arr, idx+1); | ←
}
```

$$idx = 8 \quad 4$$

5
4
0
1
3

$int n = \boxed{idx} + 1$





5 0 2 -3 2
0 1 2 3 4

```
public static void displayArrReverse(int[] arr, int idx) {  
    if(arr.length == idx) return;  
    → displayArrReverse(arr, idx+1);  
    pos1 System.out.println(arr[idx]);  
}
```

2 ✓
-3 ✓
2
0
5

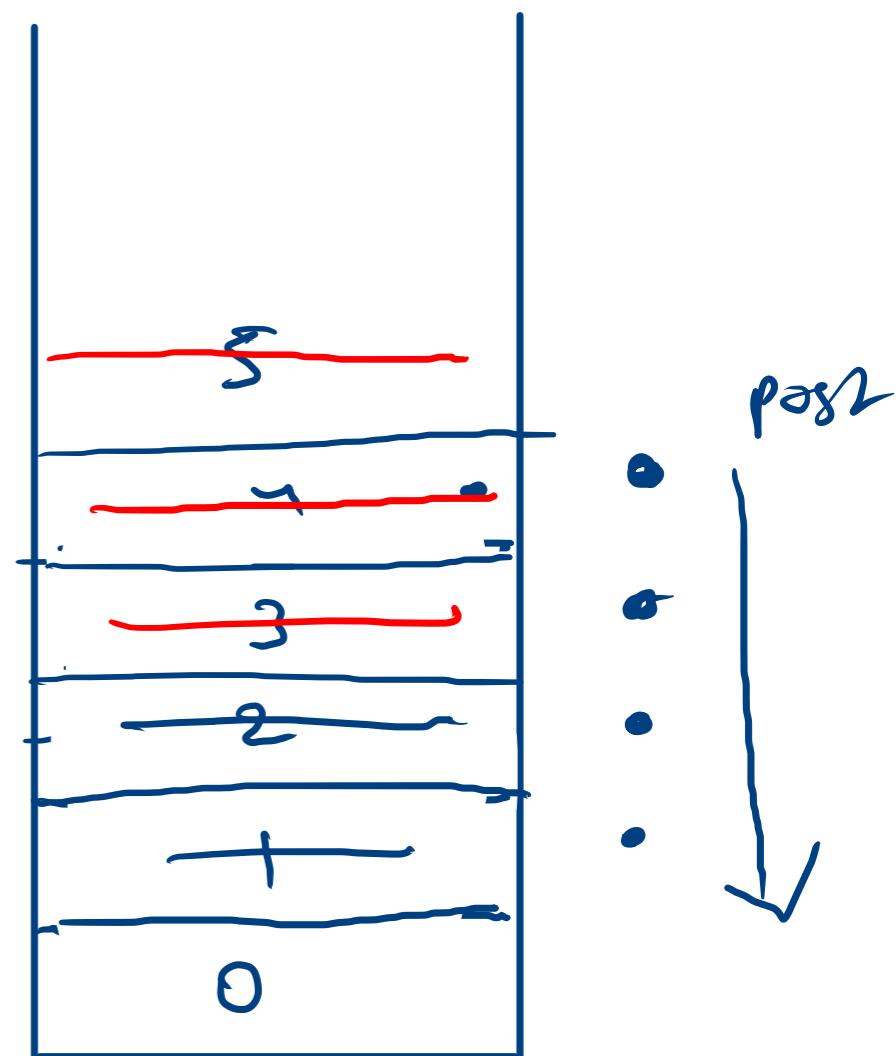
```
public static void displayArr(int[] arr, int idx){  
    if(idx == arr.length){  
        return;  
    }  
    pr → System.out.println(arr[idx]);  
    displayArr(arr, idx+1);  
    idx+1
```

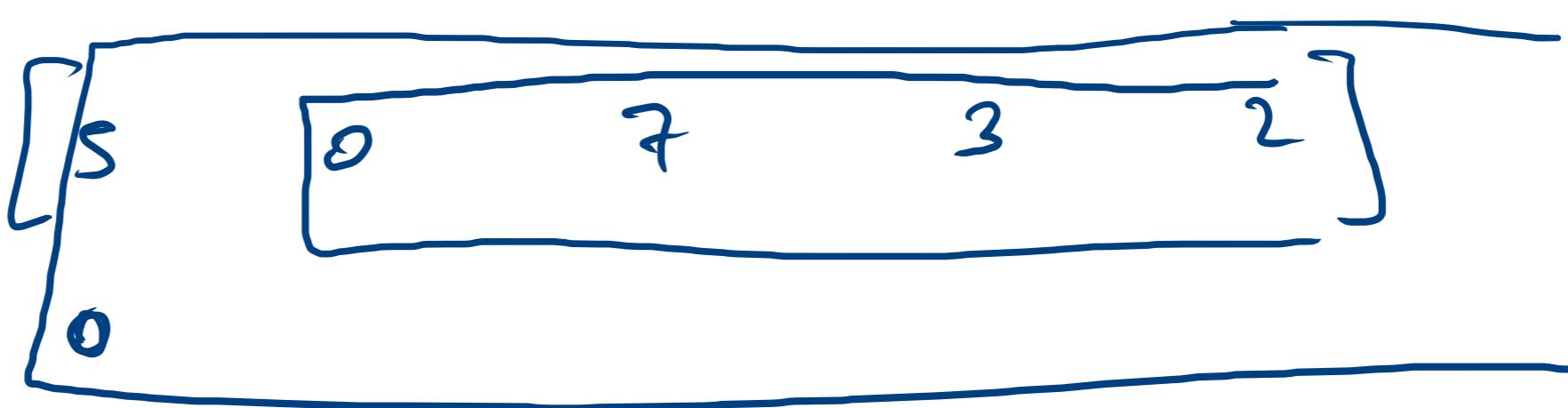
5 ✓
0 ✓
2 ✓
-3
2

pre arr -

[full()]

post arr]





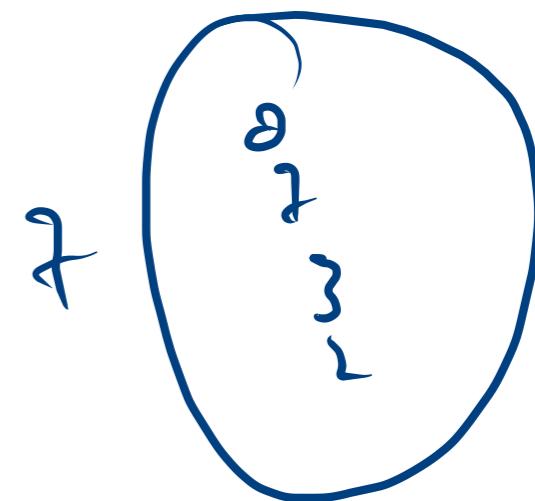
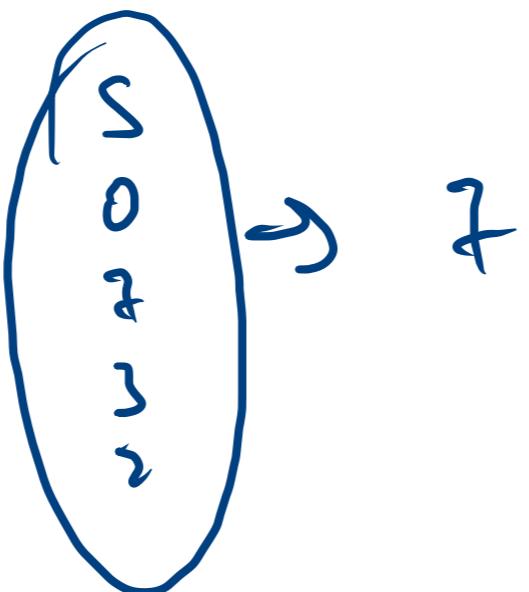
expr

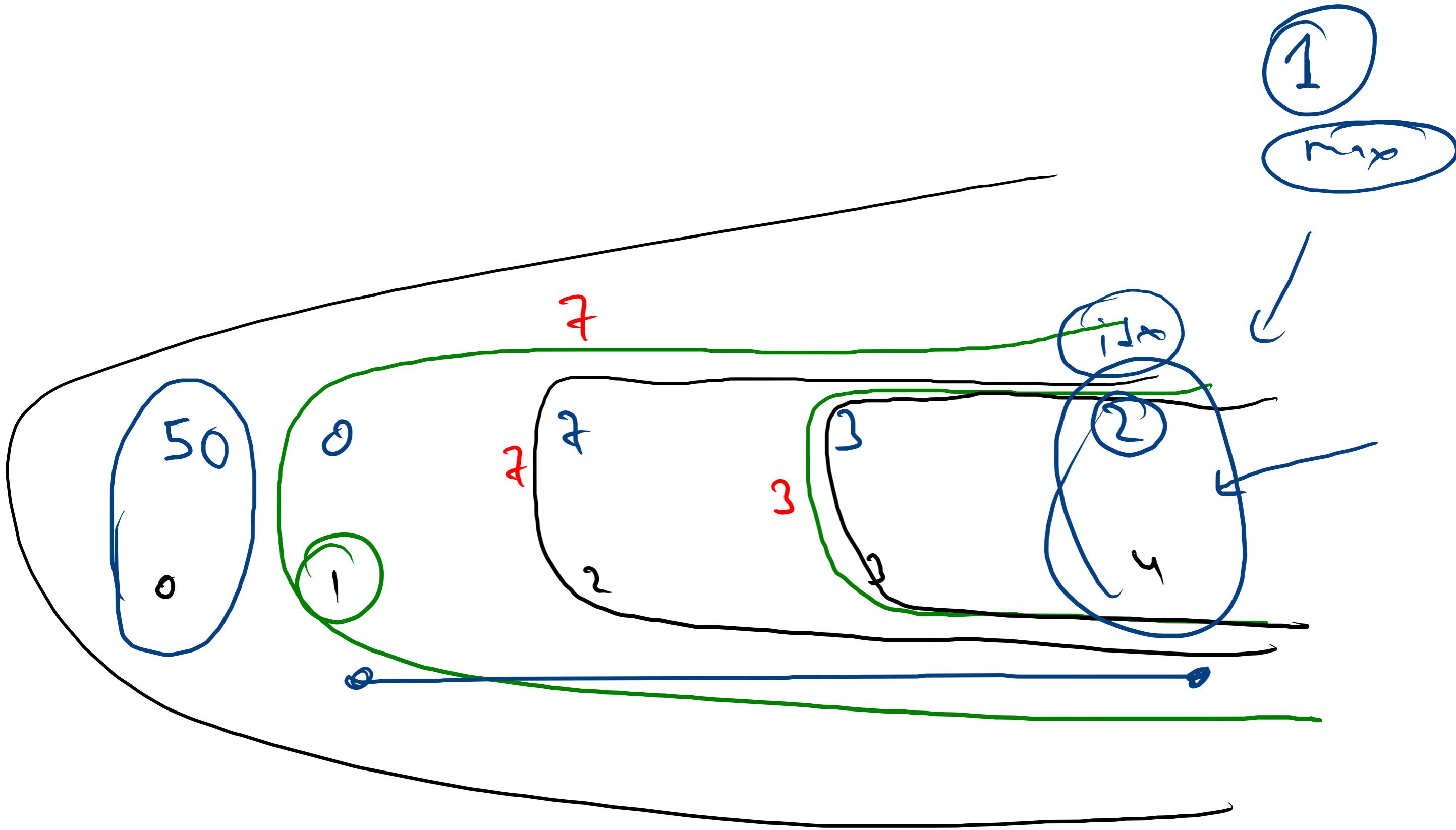
$$idx = 0$$

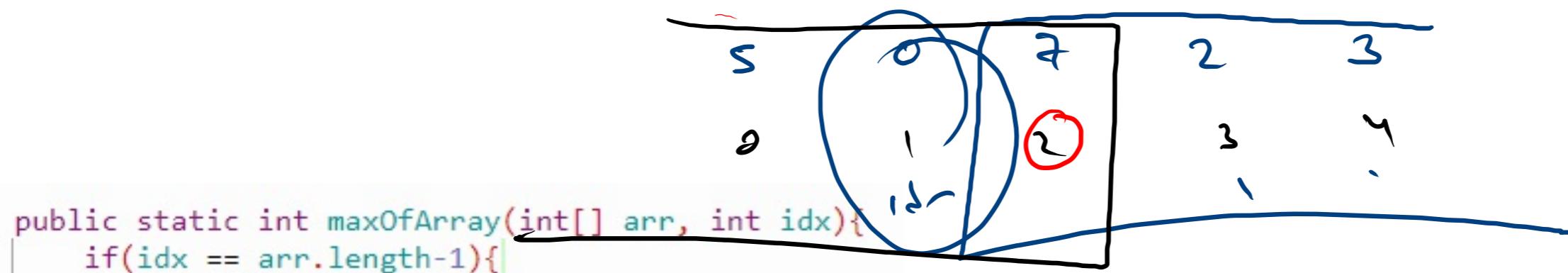
call

$$idx = 1$$

alpha \rightarrow 7







```

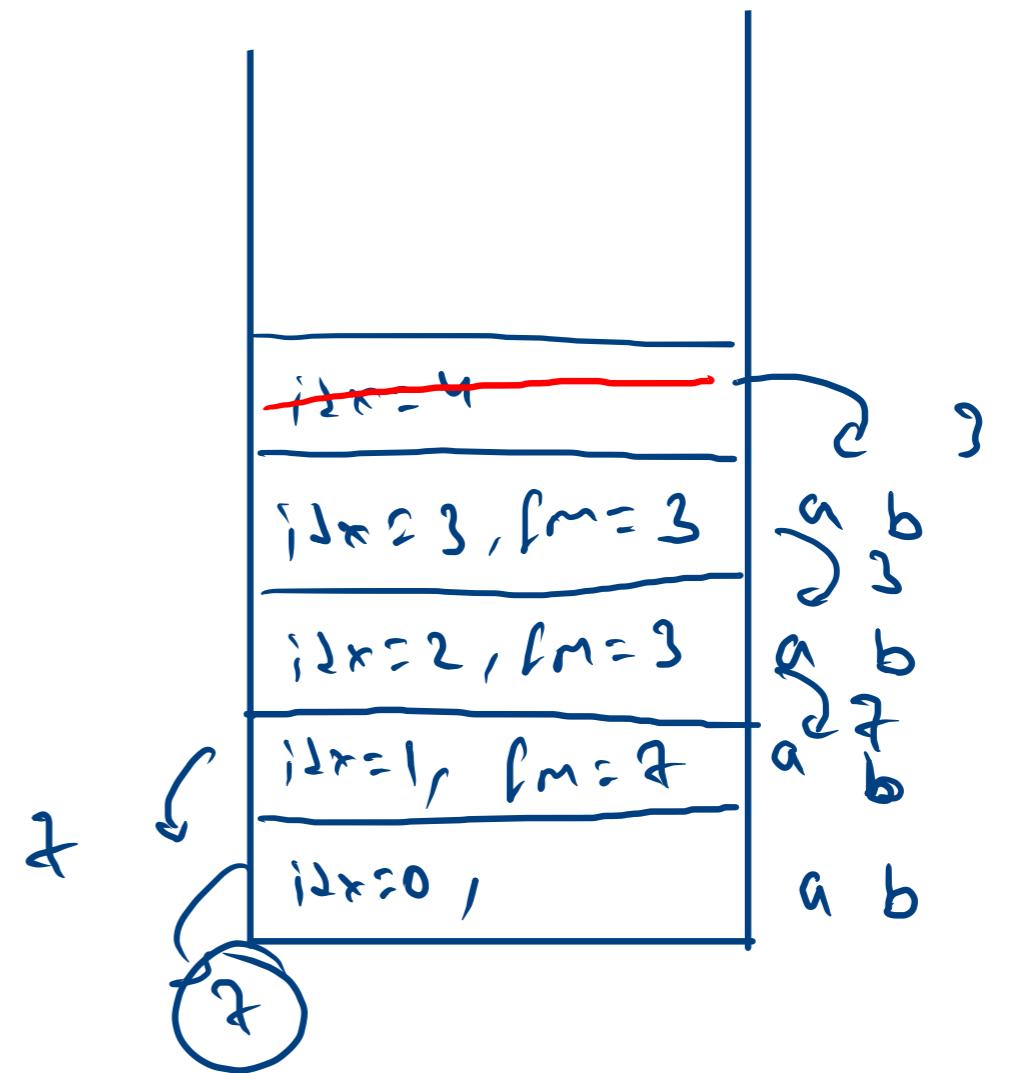
public static int maxOfArray(int[] arr, int idx){
    if(idx == arr.length-1){
        return arr[idx];
    }

    int faithmax = maxOfArray(arr, idx+1); 7 9

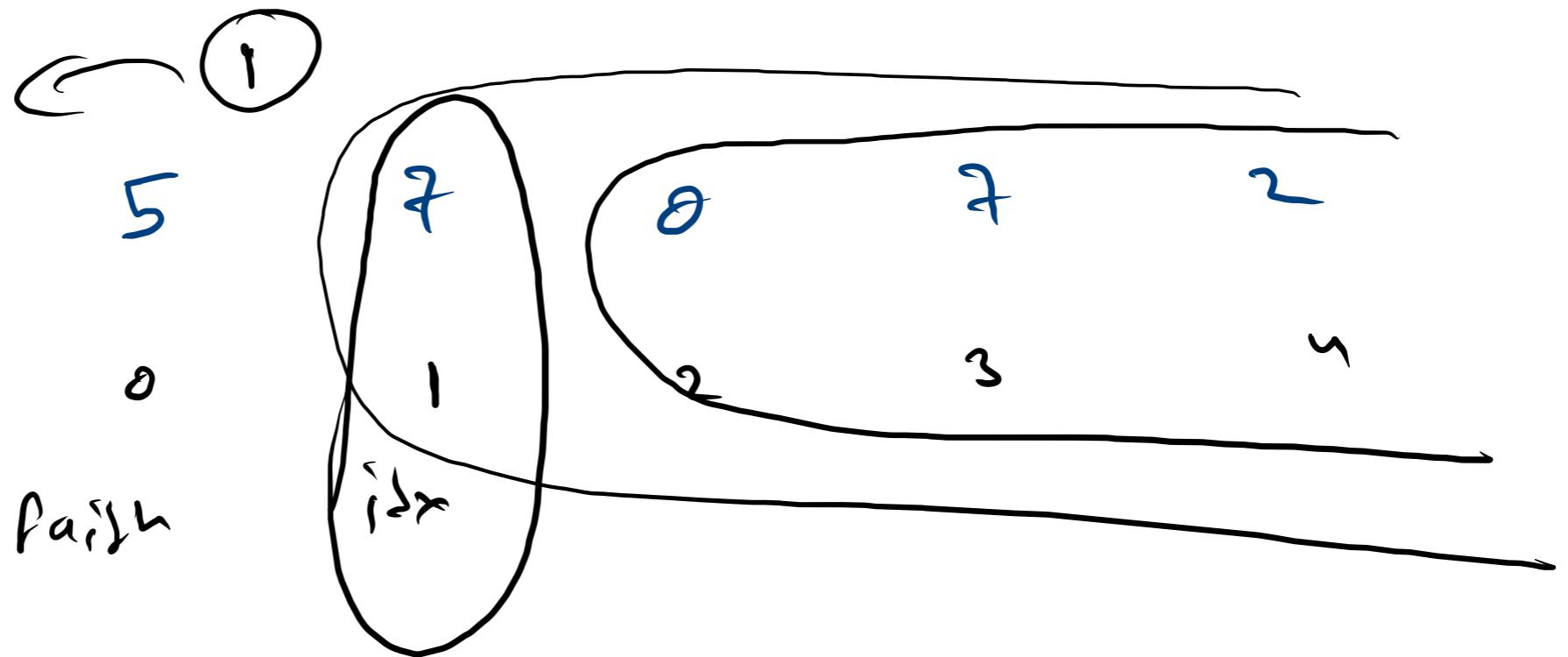
    if(arr[idx] > faithmax){ b
        return arr[idx];
    }else{ a
        return faithmax;
    }
}
  
```

exp

$$\text{idx} = 2$$



first



$n \rightarrow 7$

1 ↙

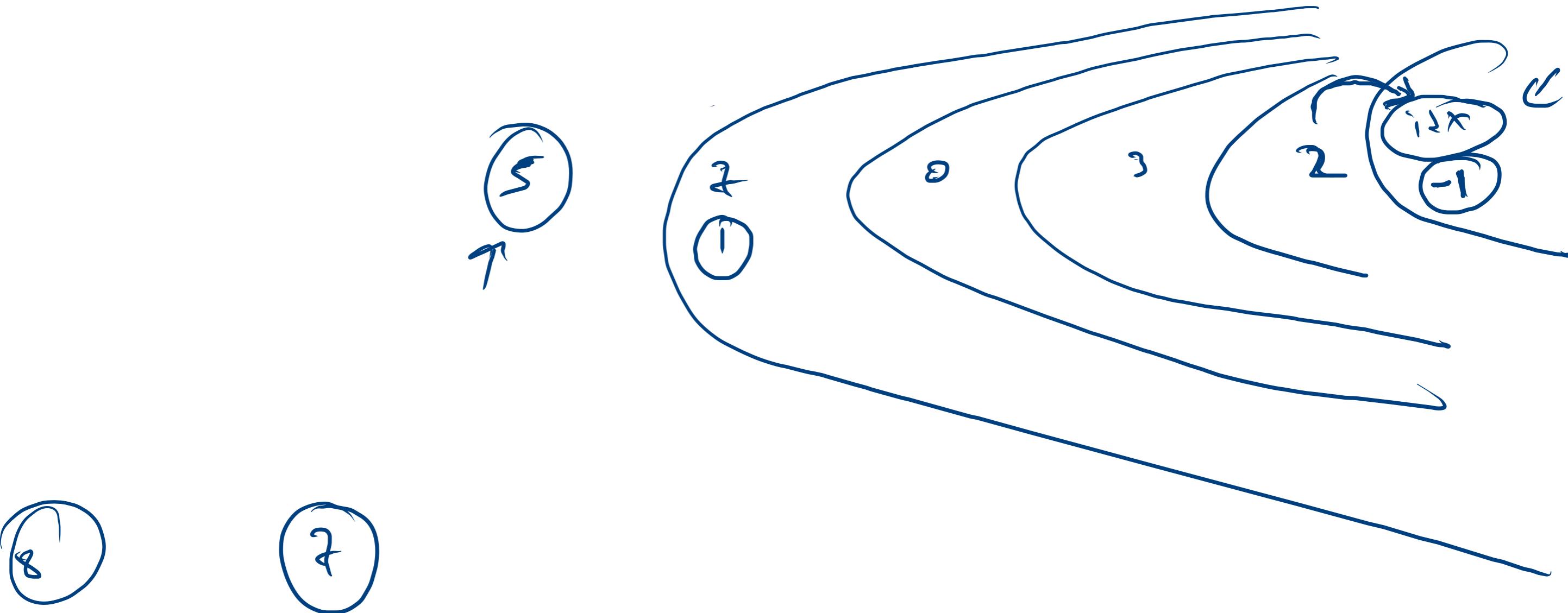
$n \rightarrow 8$

-1 ↙

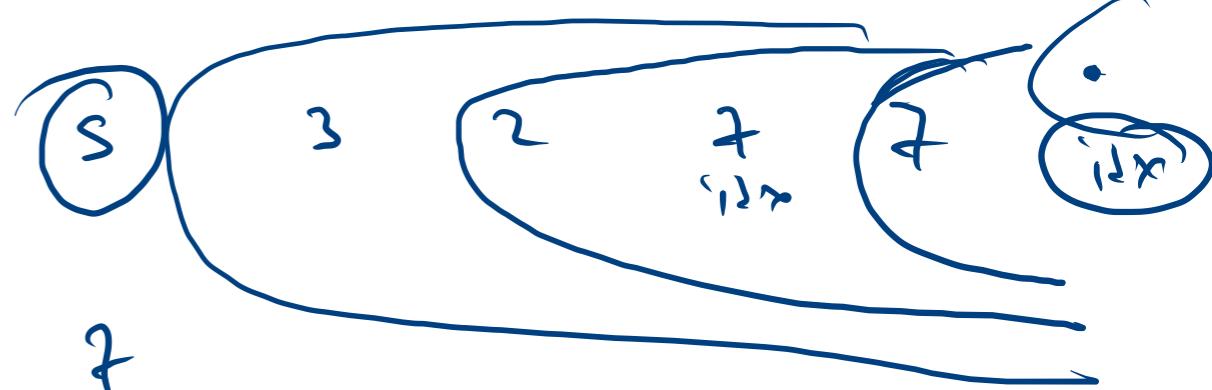
$n = 7$

pre

$\{idx\} \subseteq n$
role idx



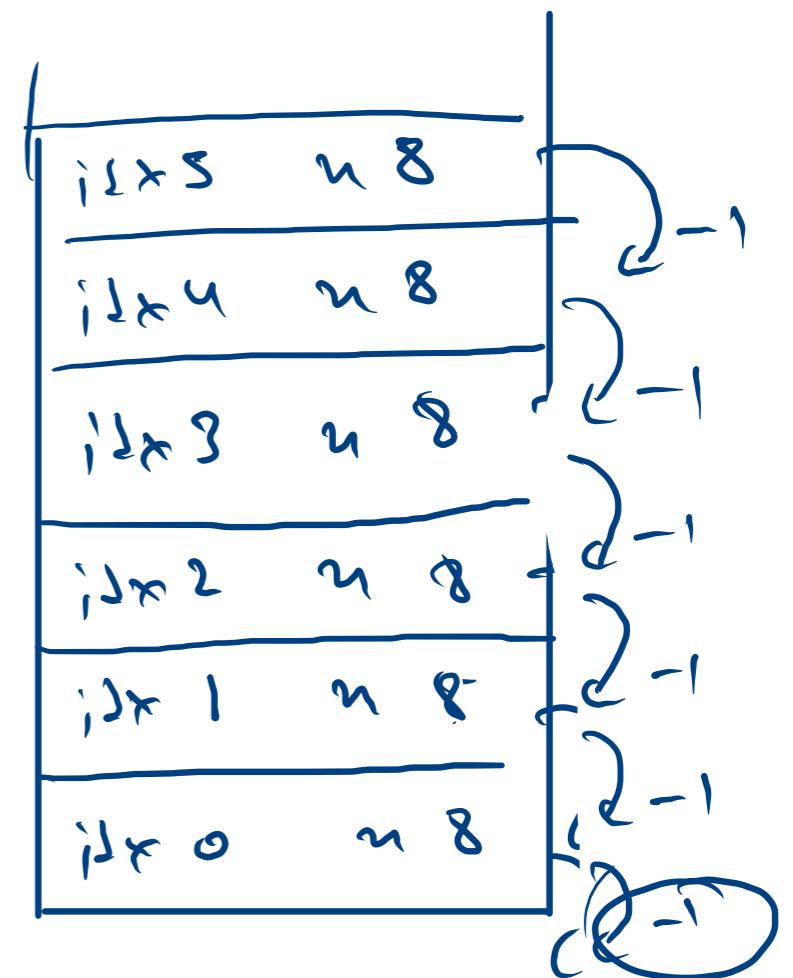
$n = 7$



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

    if(arr[idx] == x){ } } pre
        return idx;
    }

    return firstIndex(arr, idx+1, x);
}
```



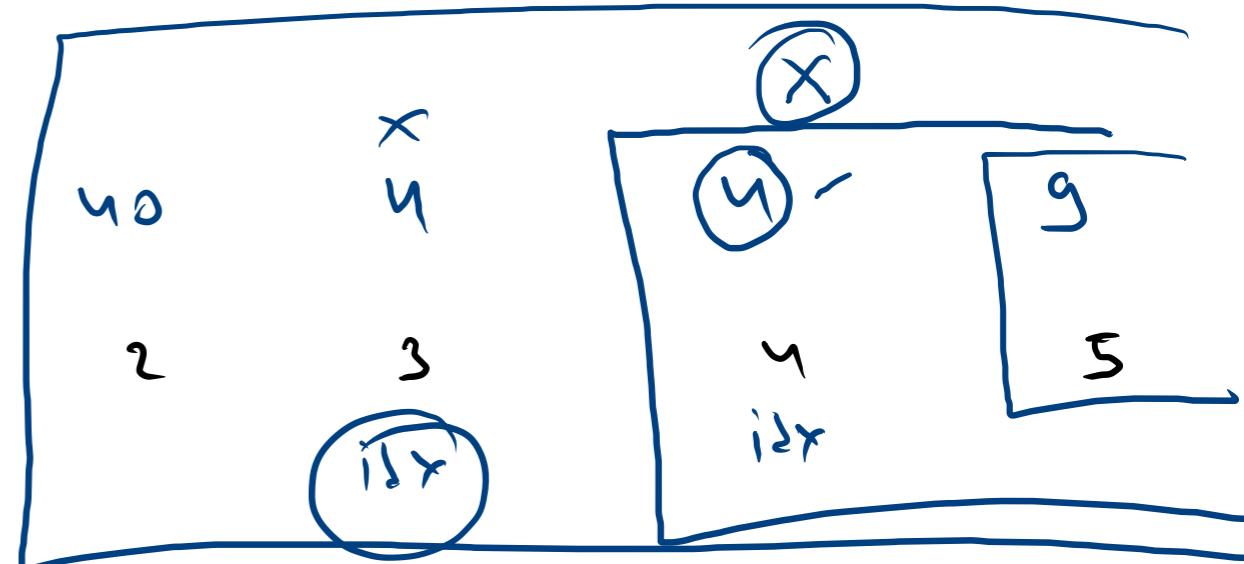
15 11 40 4 4 9

4

$x = 4$

Last

15
0
1
 idx

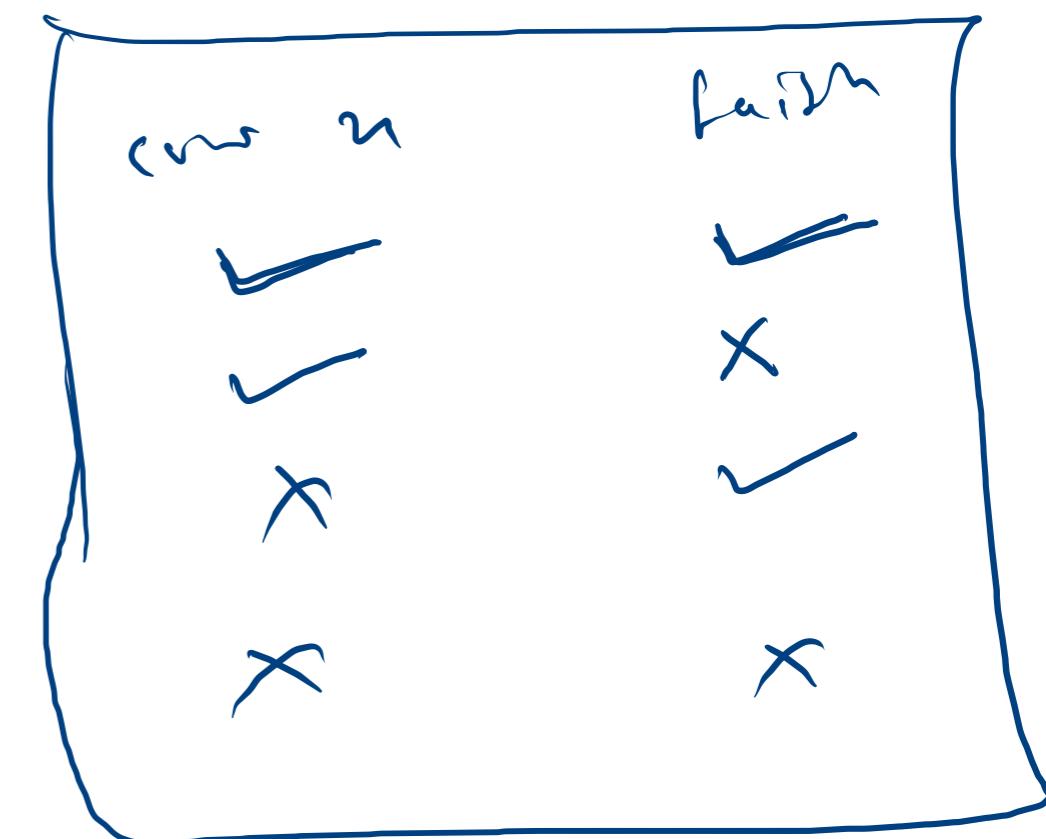


expectation

$idx = 1$

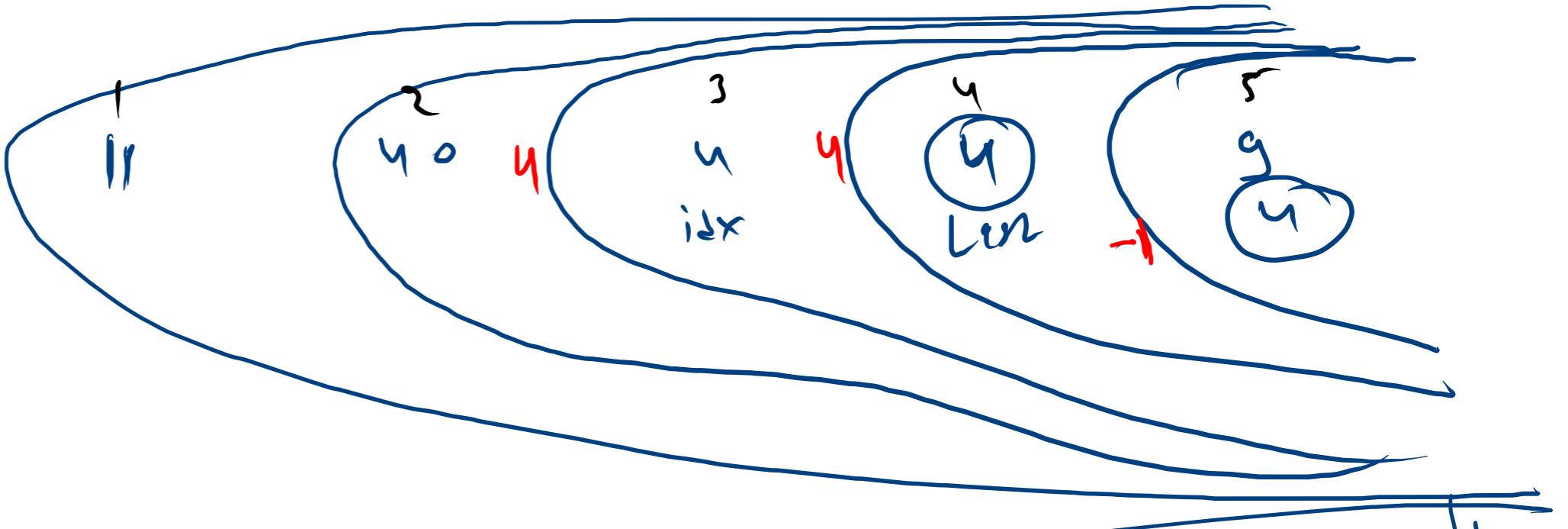
ans = 4

$x = 8$



Las2

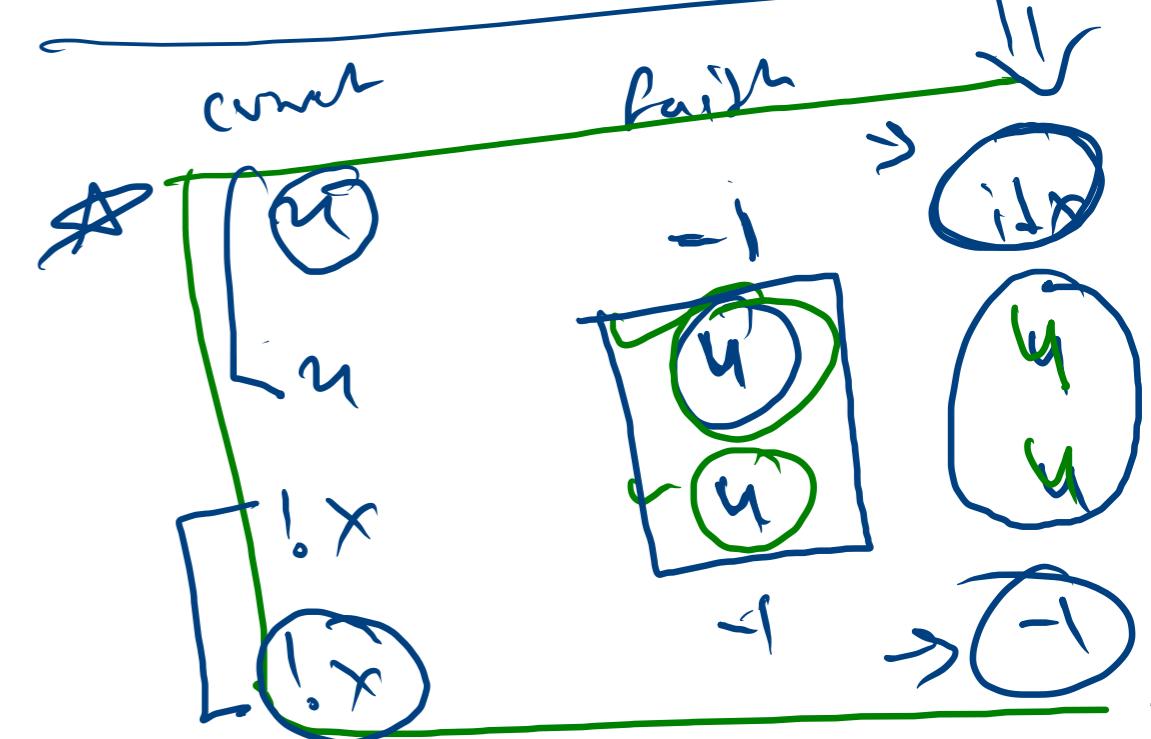
0
15
 i_{2x}



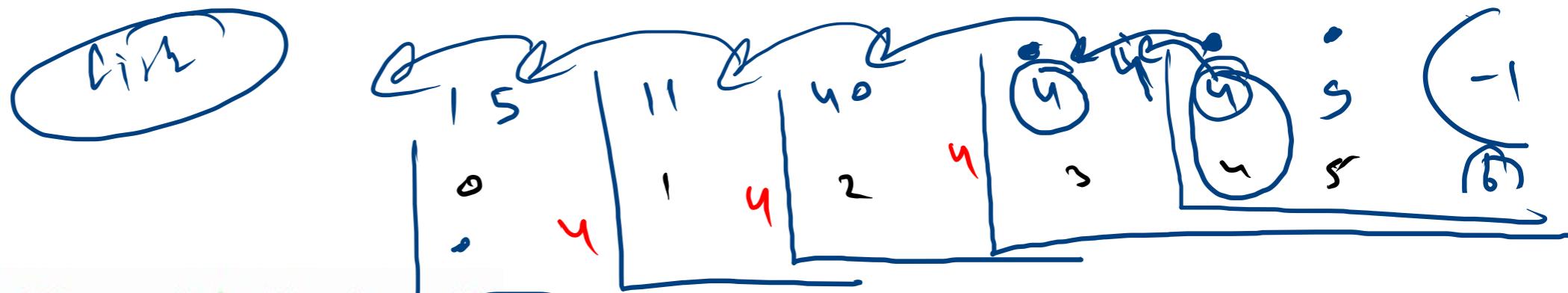
$x = 4$

$$fI = \text{failm}(i_{2x} + 1)$$

$\text{if } (fI != -1)$
return fI
 $\text{if } (= -n)$



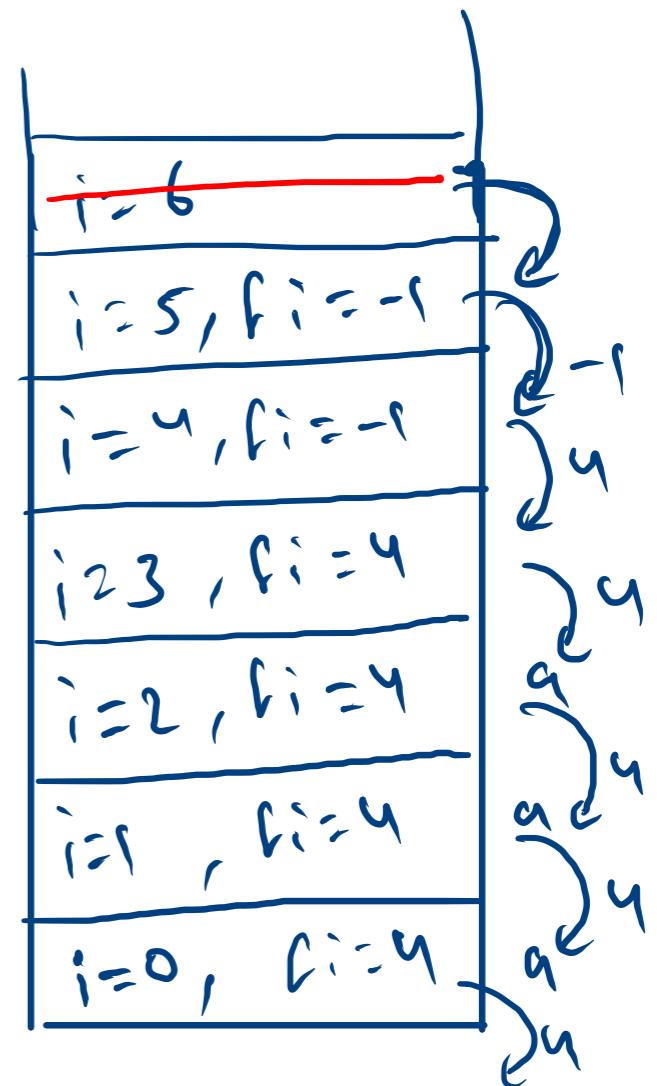
15 11 40 4 4 9
4

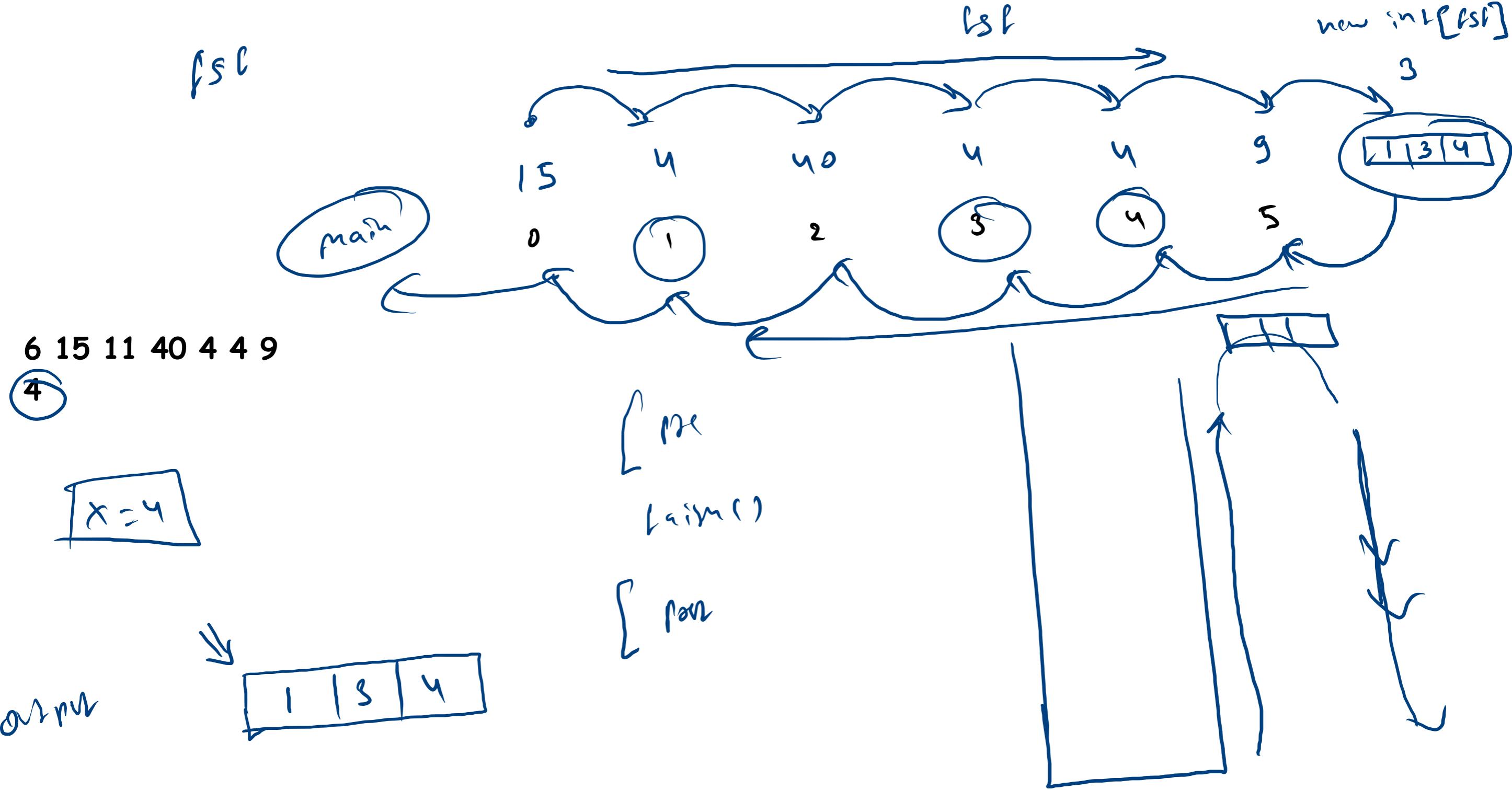


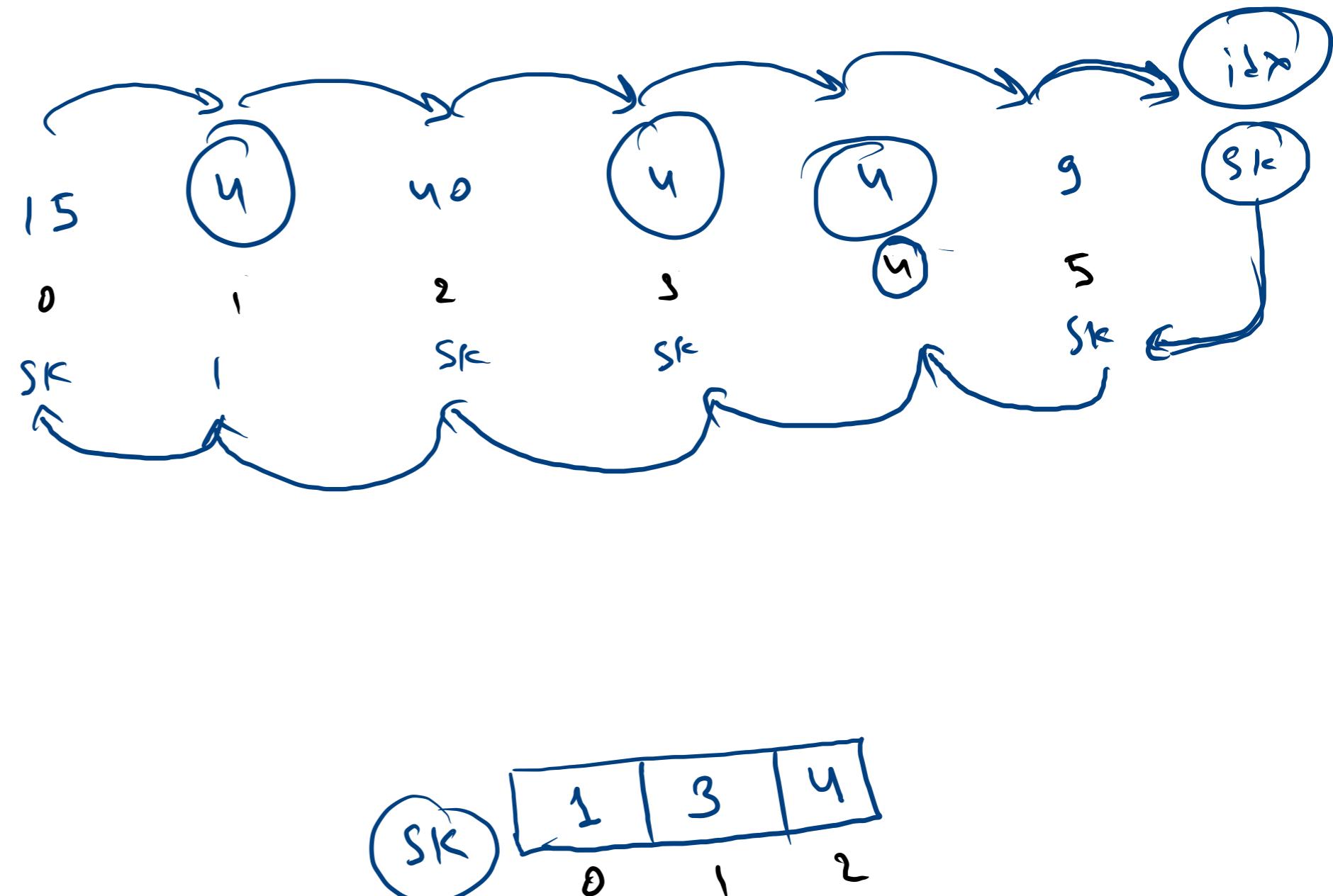
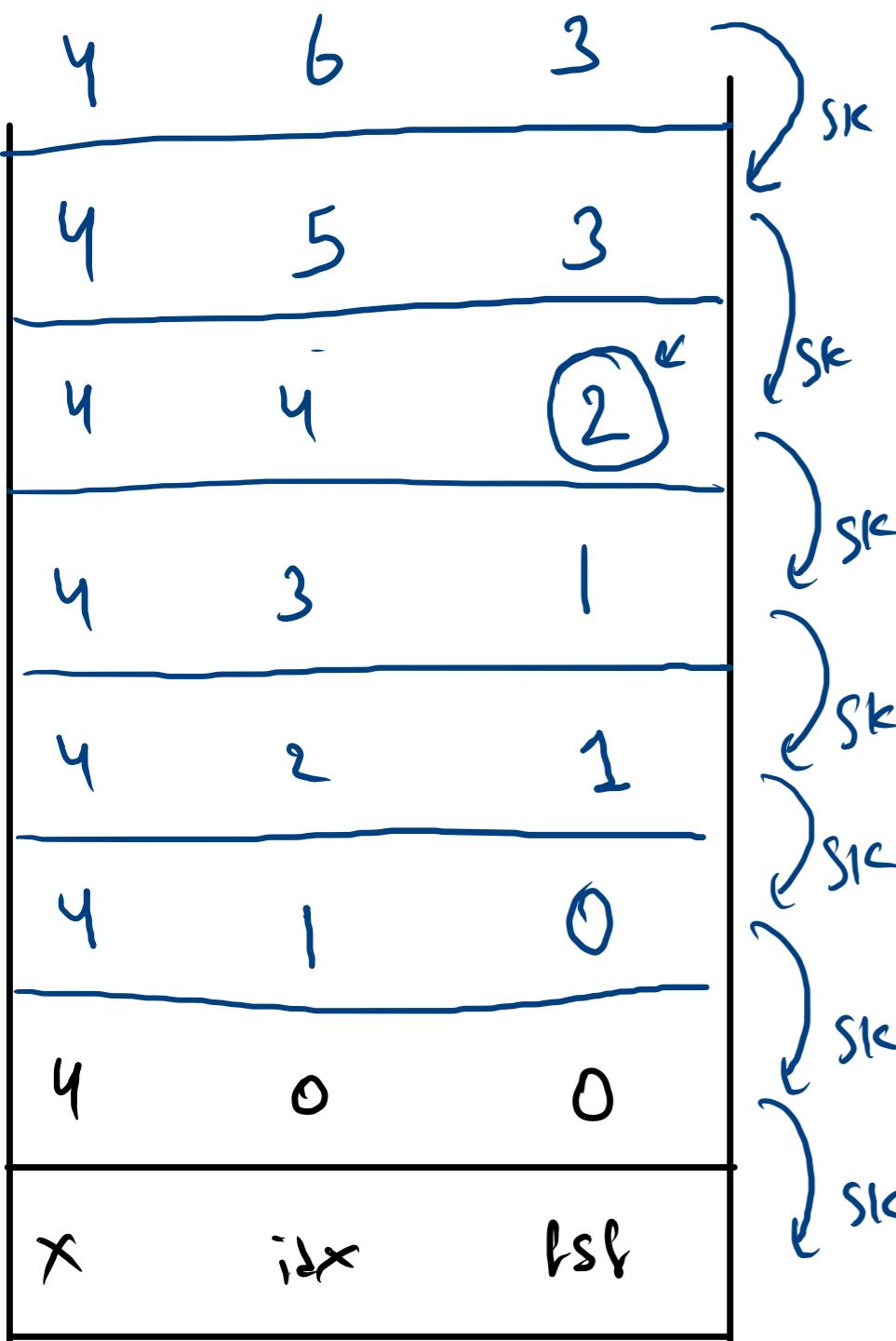
```
public static int lastIndex(int[] arr, int idx, int x){  
    if(arr.length == idx) return -1;  
  
    int faithIndex = lastIndex(arr, idx+1, x); a  
    if(faithIndex != -1){  
        return faithIndex;  
  
    if(arr[idx] == x) return idx;  
  
    return -1;  
}
```

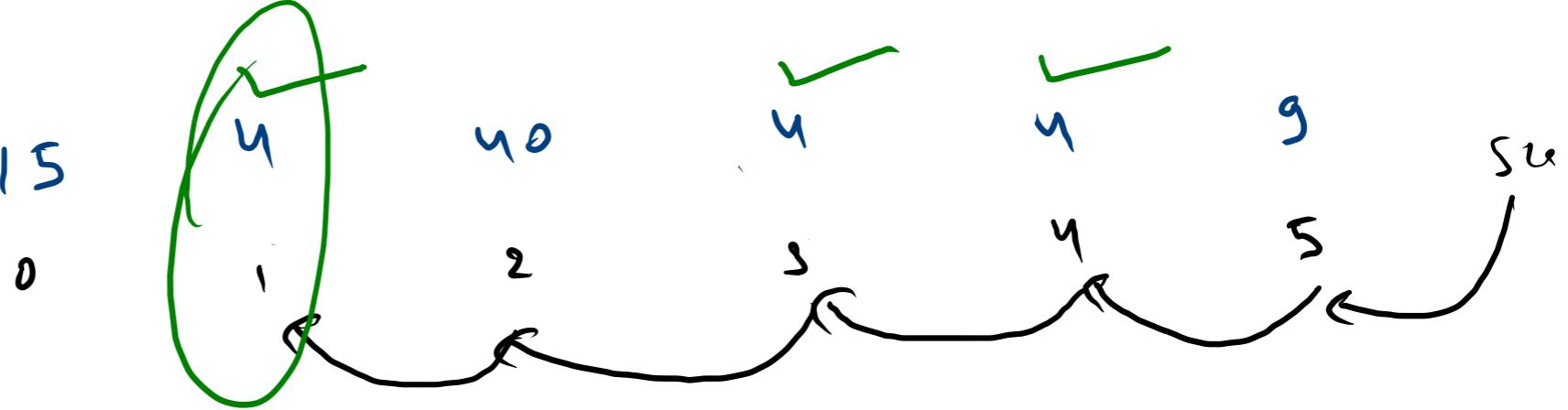
st

par









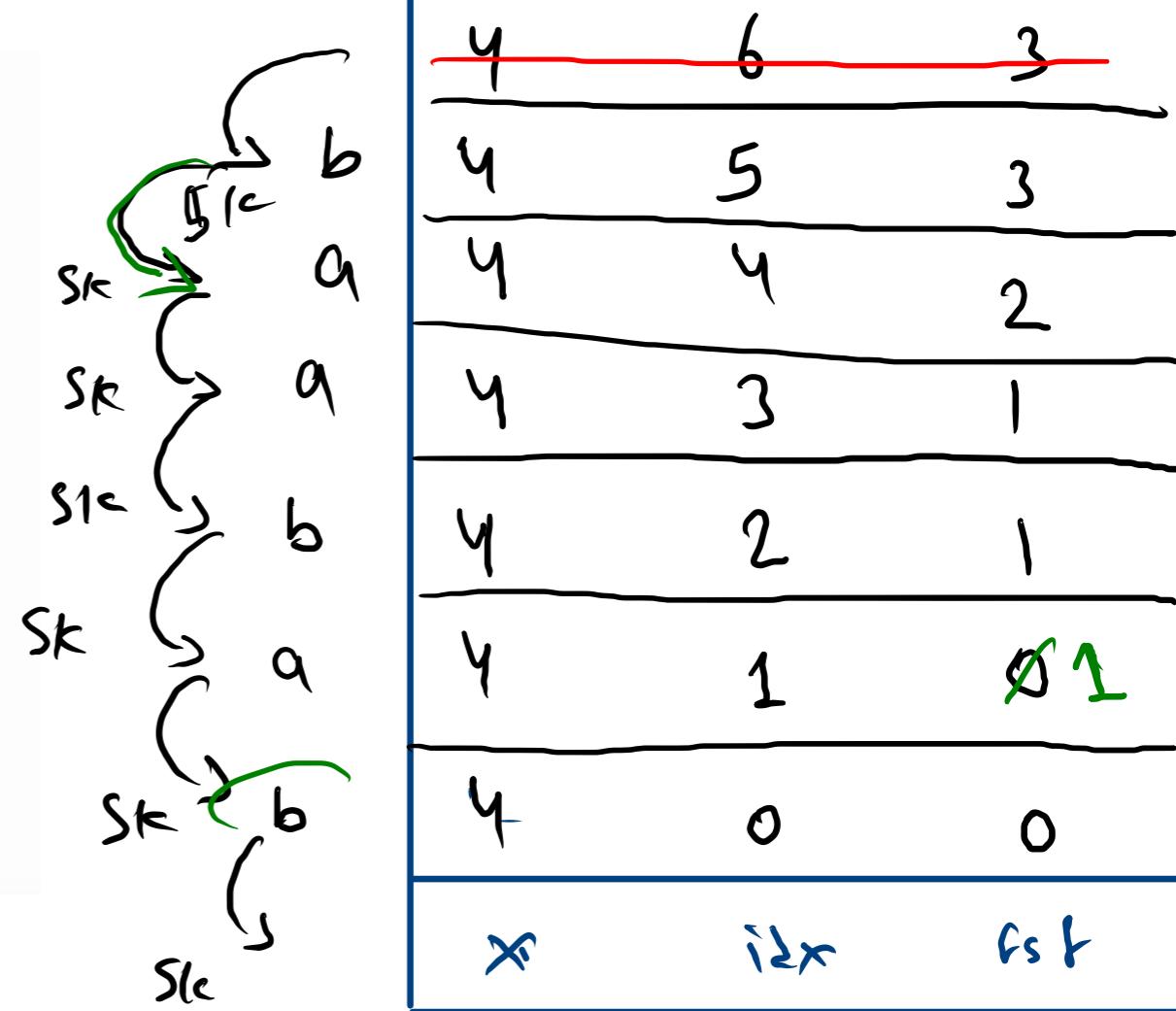
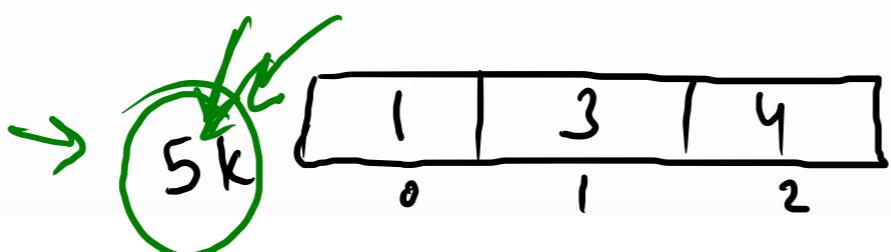
```

public static int[] allIndices(int[] arr, int x, int idx, int fsf) {
    if(arr.length == idx){
        int ar[] = new int[fsf];
        return ar;
    }

    if(arr[idx] == x){
        int[] iarr = allIndices(arr, x, idx+1, fsf+1);
        iarr[fsf] = idx; ←
        return iarr;
    }else{
        int[] iarr = allIndices(arr, x, idx+1, fsf);
        return iarr;
    }
}

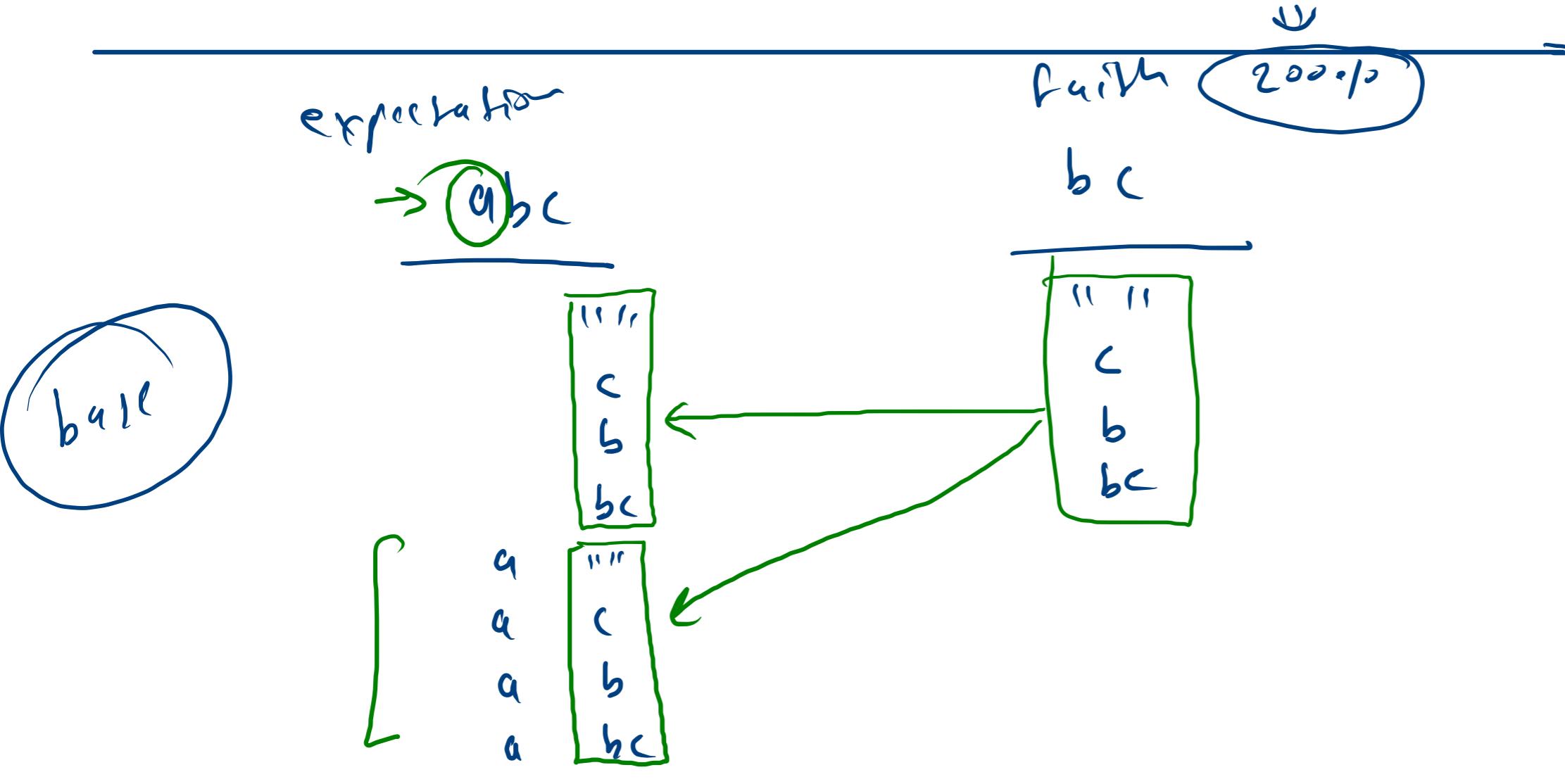
```

lsl-1



$s \rightarrow abc$

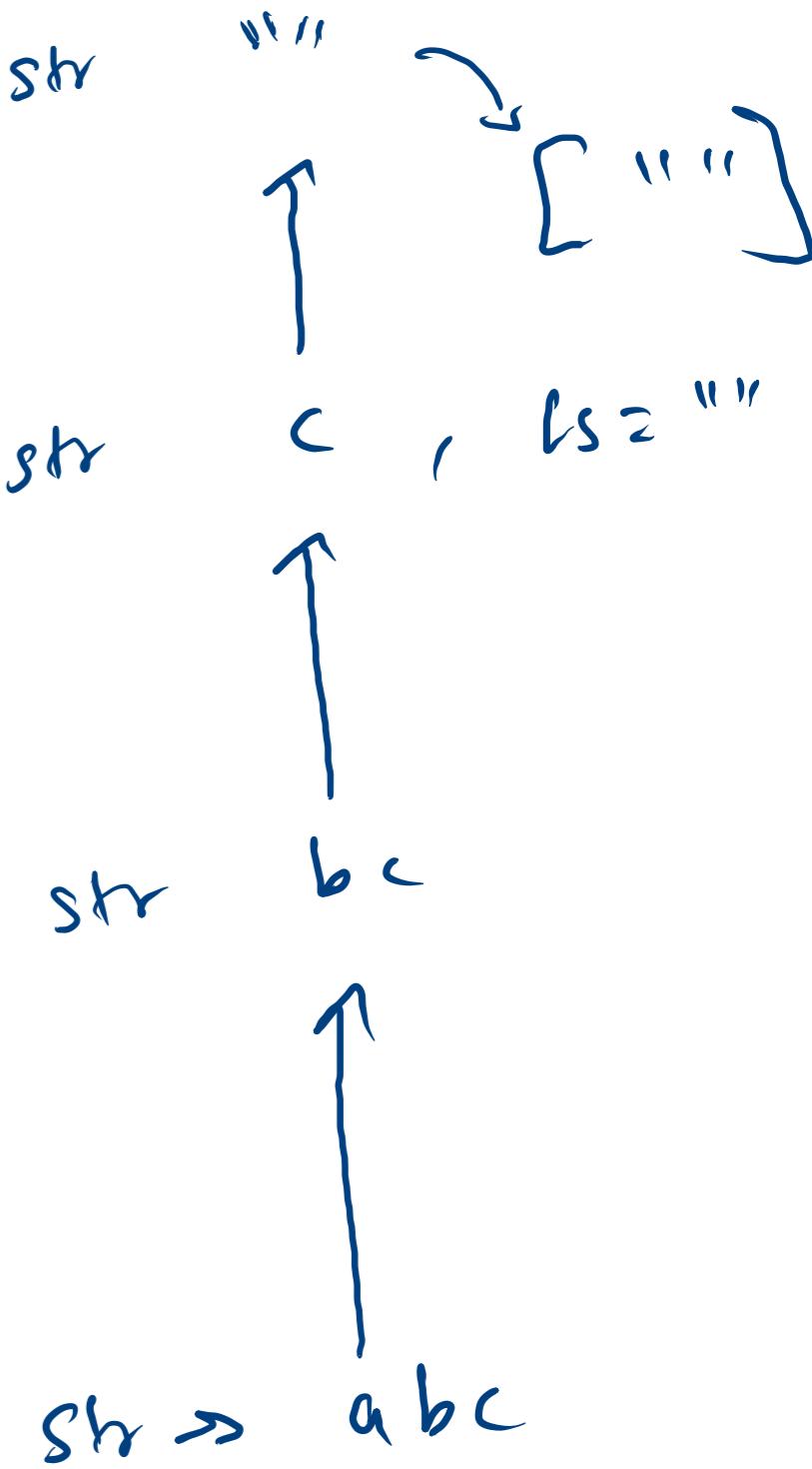
ALIZ $\Rightarrow \{ "", c, b, bc, a, ac, ab, abc \}$



abc
substituy (1)

i
id + e1

```
    "" → [ "" ]  
public static ArrayList<String> gss(String str) {  
    String fs = str.substring(1);  
    ArrayList<String> faithAl = gss(fs);  
  
    ArrayList<String> myAl = new ArrayList<>();  
  
    for(String s: faithAl){  
        myAl.add(s);  
    }  
  
    for(String s: faithAl){  
        myAl.add(str.charAt(0) + s);  
    }  
  
    return myAl;  
}
```



```

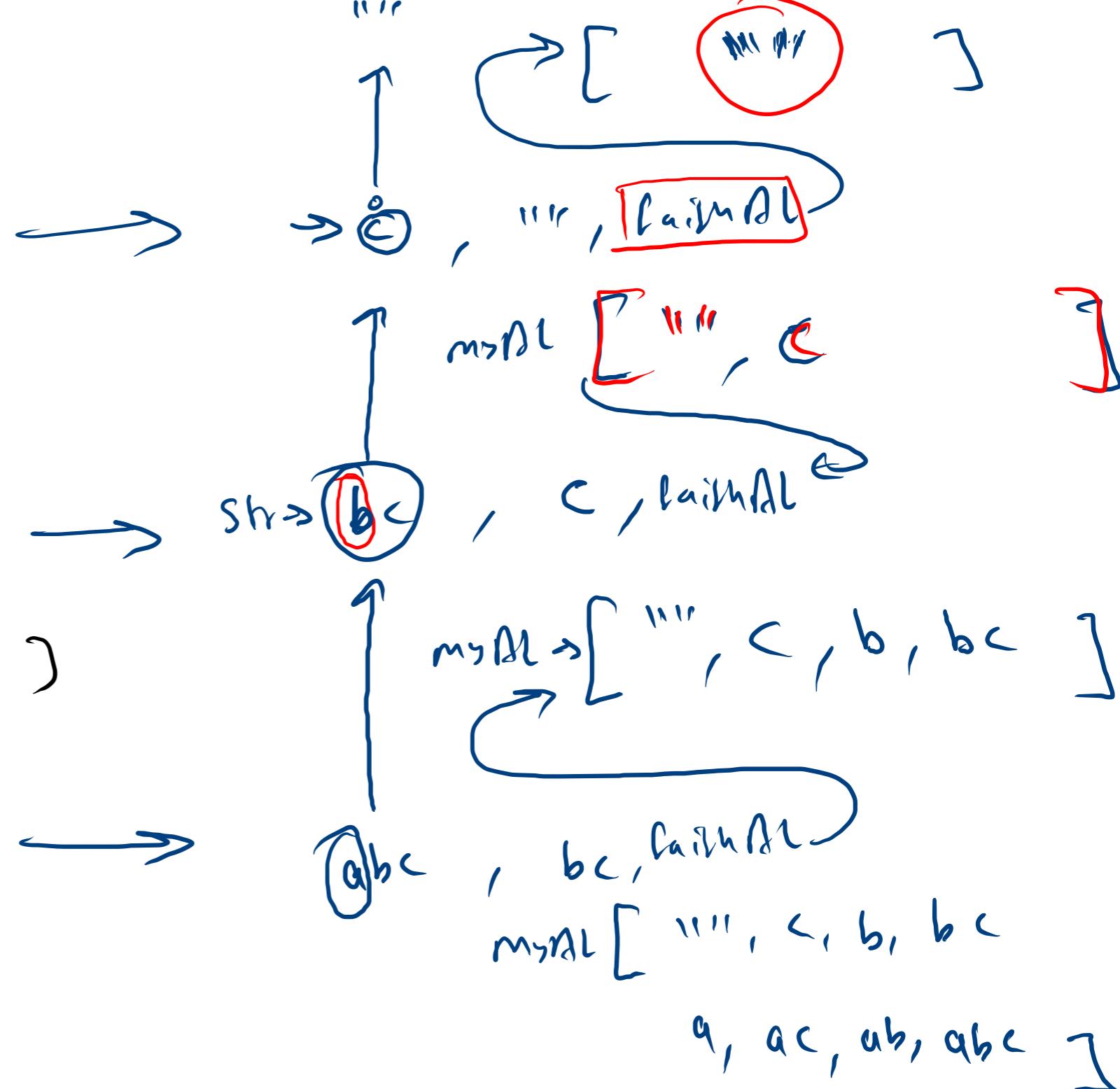
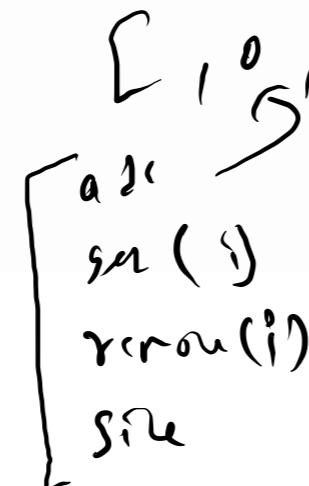
public static ArrayList<String> gss(String str) {
    if(str.length() == 0){
        ArrayList<String> myAl = new ArrayList<>();
        myAl.add("");
        return myAl;
    }
    String fs = str.substring(1);
    ArrayList<String> faithAl = gss(fs);

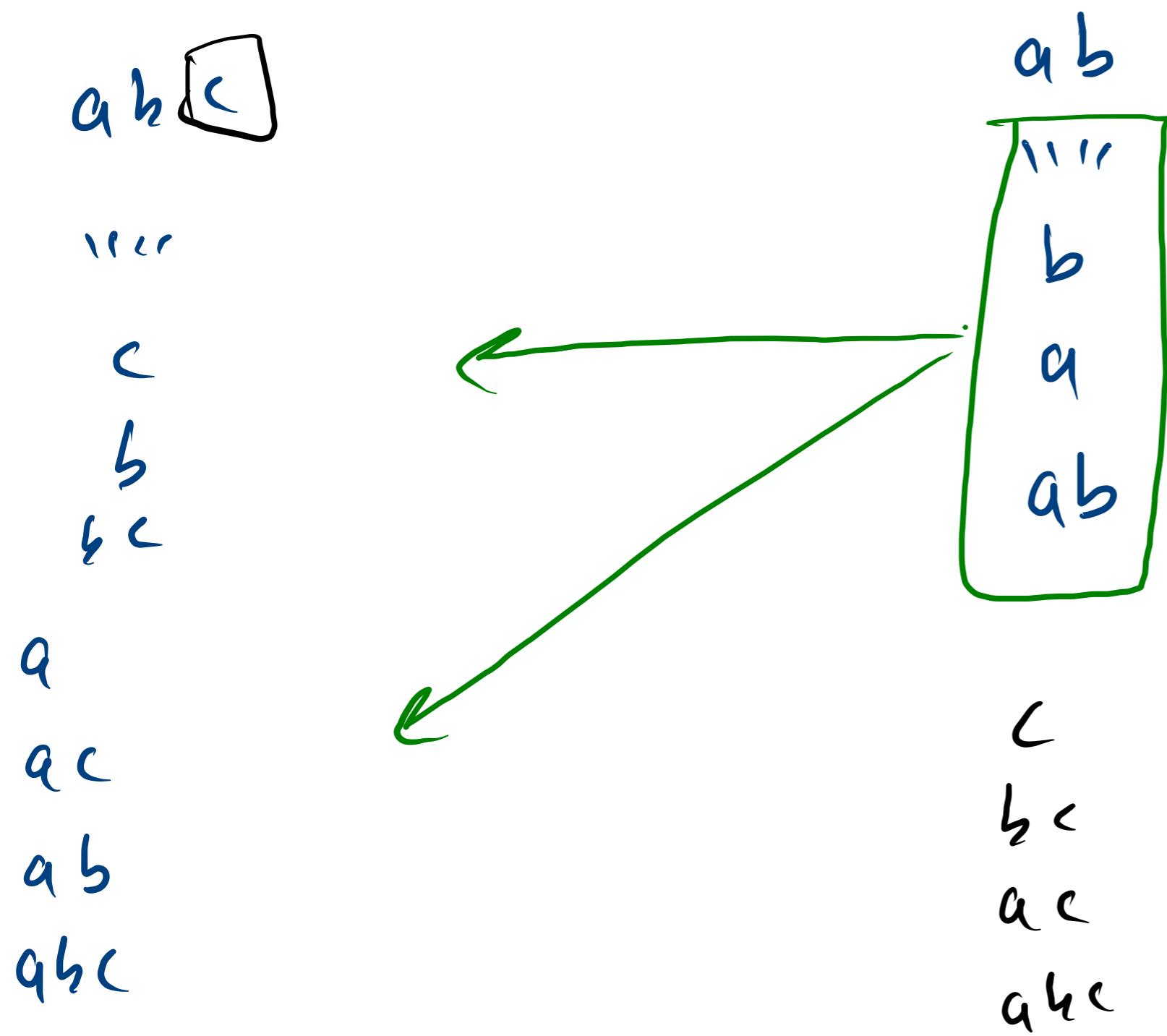
```

```

    ArrayList<String> myAl = new ArrayList<>();
    for(String s: faithAl){
        myAl.add(s);
    }
    for(String s: faithAl){
        myAl.add(str.charAt(0)+ s);
    }
    return myAl;
}

```





7 9

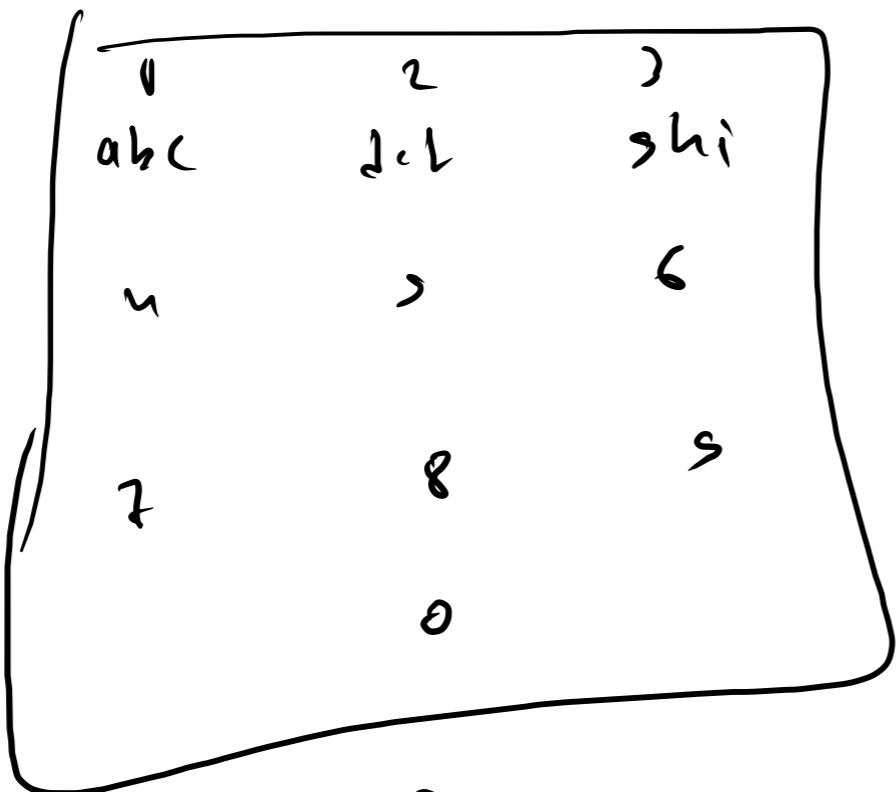
t y	v y
t z	v z

"2"

[d, e, f]

"2 3"

d g d h d i	j g / e h / e i	l g f h l i
c g / e h / e i	f g f h f i	r g r h r i



2 3
d g j c f r
e f i h h i
h h i l l i
d d i l l i
l l i l l i

2 9
t y
v z

- 0 -> ;
- 1 -> abc
- 2 -> def
- 3 -> ghi
- 4 -> jkl
- 5 -> mno
- 6 -> pqrs
- 7 -> tu
- 8 -> vwx
- 9 -> yz

```
public static void pzz(int n){  
    if(n==0) return;  
    System.out.print(n+" ");  
    pzz(n-1);  
    System.out.print(n+" ");  
    pzz(n-1);  
    System.out.print(n+" ");  
}
```

