

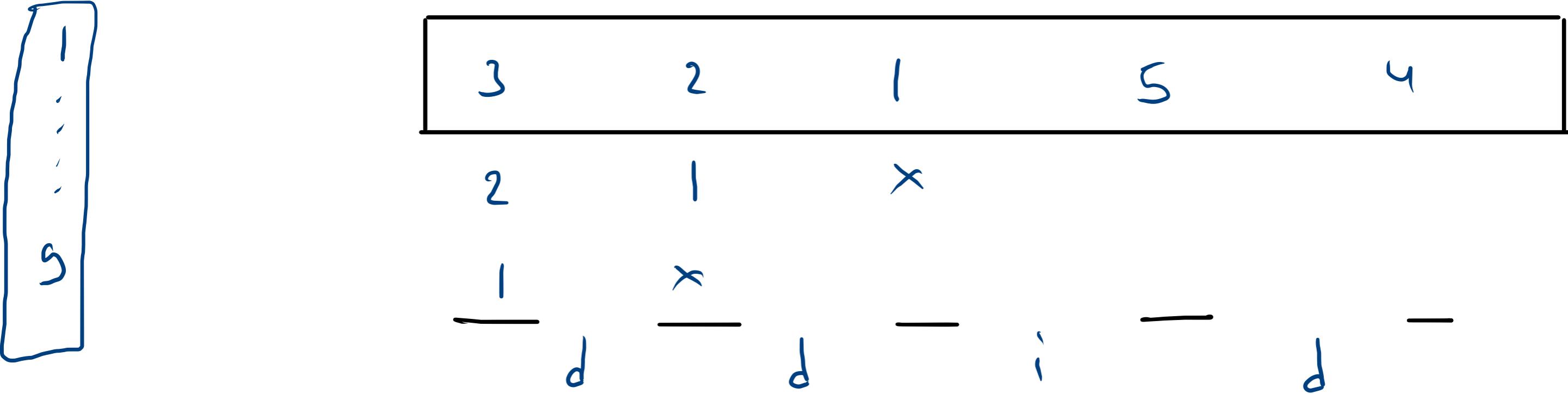
*d, i*

1...9

$$\begin{array}{r} \underline{7} & \underline{6} & \underline{4} & \underline{5} & \underline{9} \\ d & d & i & i & \end{array}$$

$$\begin{array}{r} 3 & 2 & 1 \\ \hline 9 & 8 & + \\ \hline d & d & \end{array} \quad \leftarrow$$

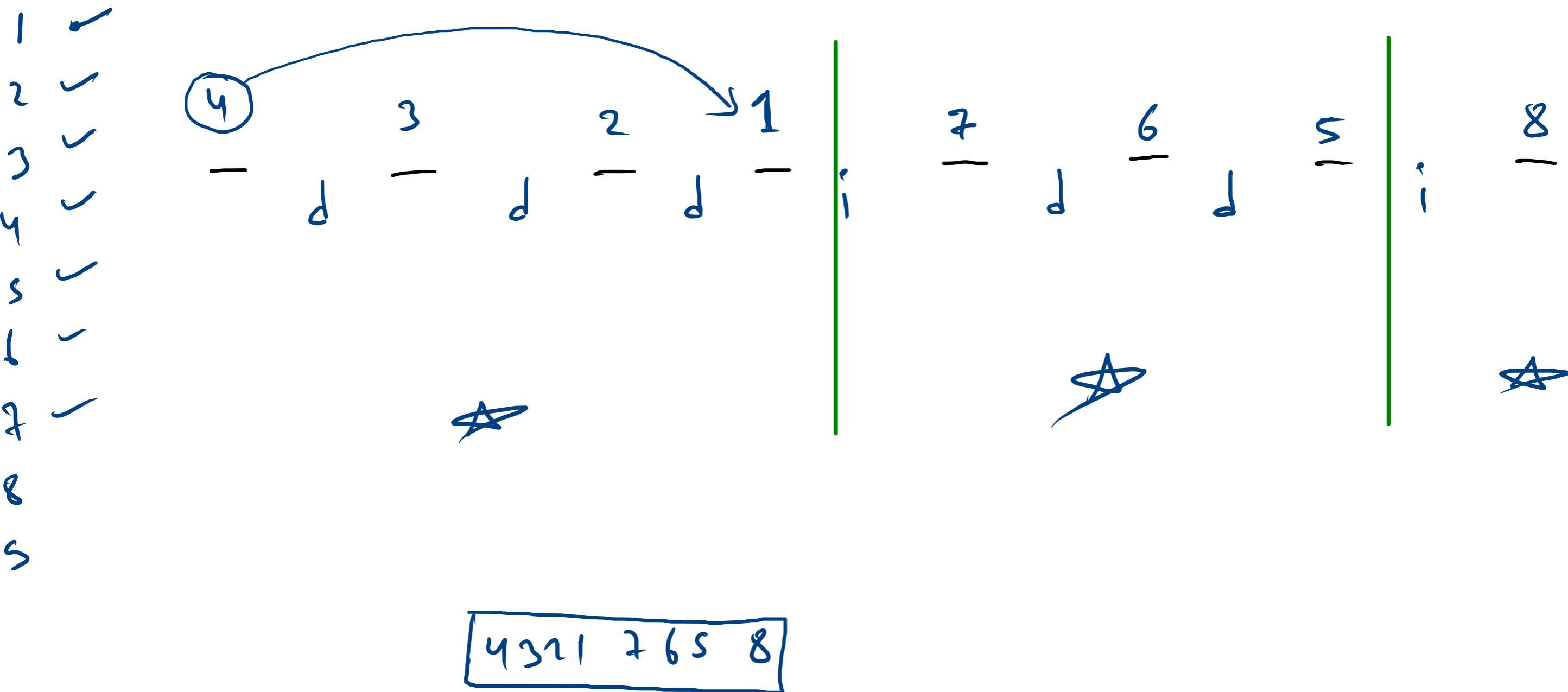
1  
-  
3  
9

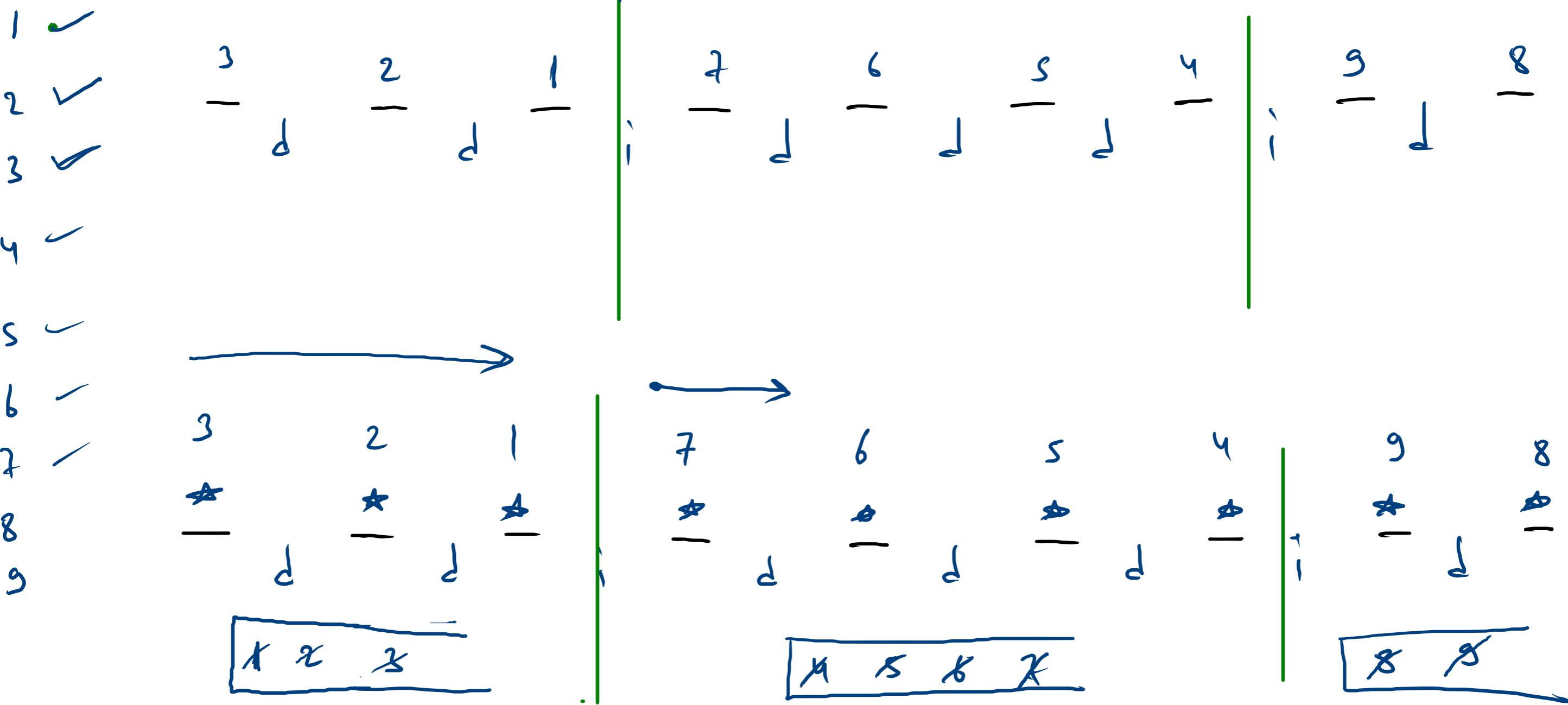


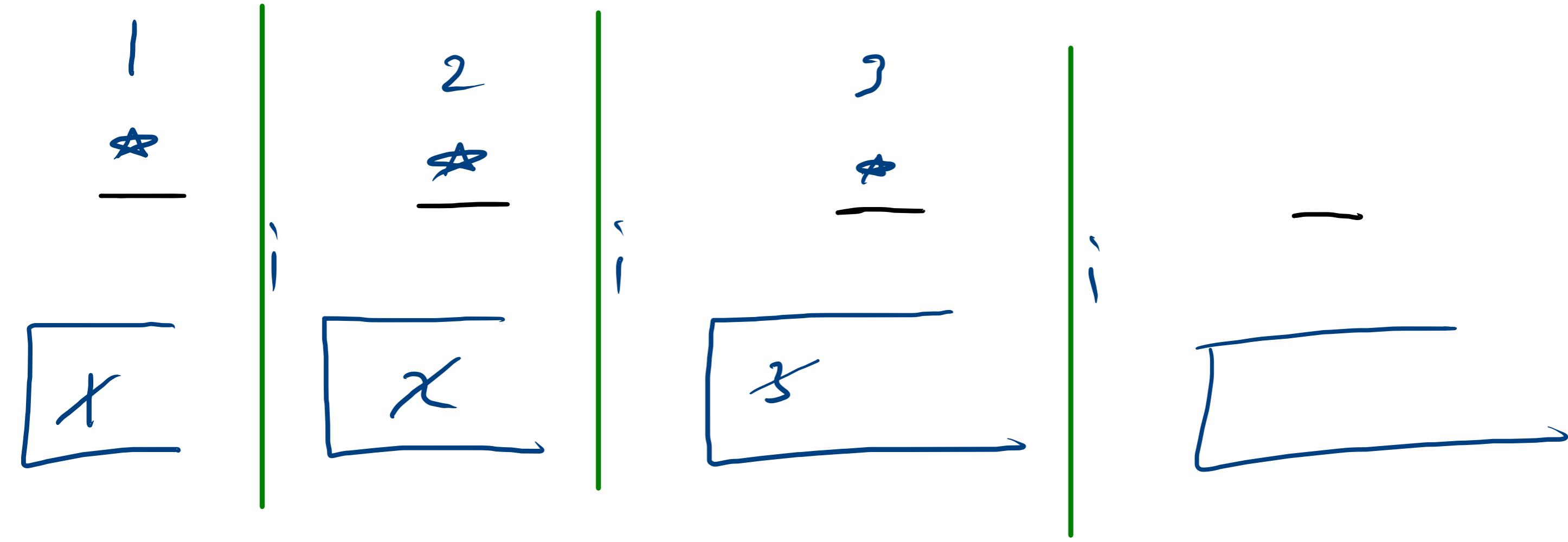
A hand-drawn diagram showing two rows of symbols. The first row contains two pairs of symbols: a dash over a dot and a dot over a dash, both with an 'x' written below them. The second row contains three pairs of symbols: a dash over a dot, a dot over a dash, and a dash over a dash, all with an 'x' written below them. The entire diagram is enclosed in a large oval.

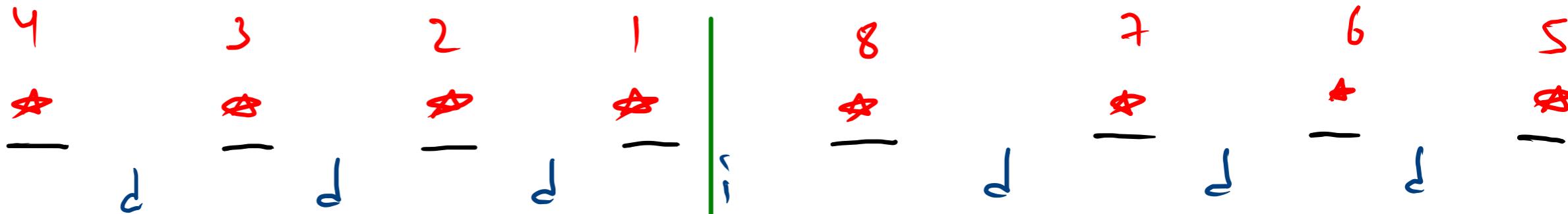
1 4 3 2 1 5 8 7 6 9  
2 1 2 1 1 1 1 1 1 1  
3 1 1 1 1 1 1 1 1 1  
4 1 1 1 1 1 1 1 1 1  
5 1 1 1 1 1 1 1 1 1  
6 1 1 1 1 1 1 1 1 1  
7 1 1 1 1 1 1 1 1 1  
8 1 1 1 1 1 1 1 1 1  
9 1 1 1 1 1 1 1 1 1

1 3 2 1  
2 3 2 1  
3 2 1 2 2









e.g.

d → 21

i → 12

ddd → 4321

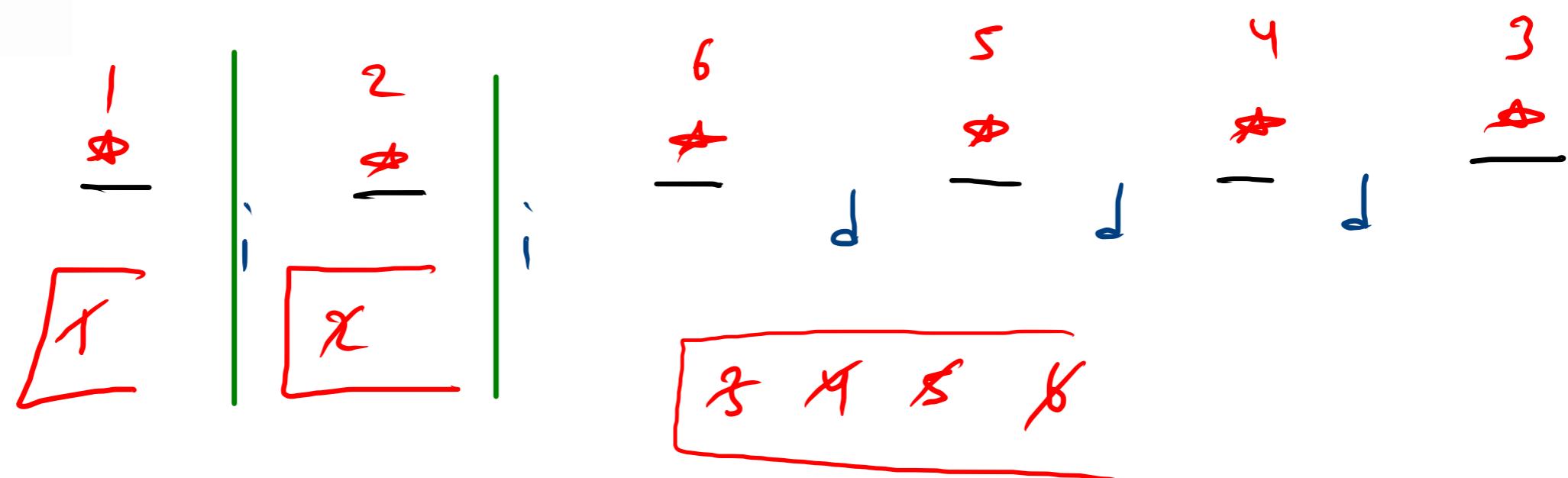
iii → 1234

ddidddd → 43218765 ↵

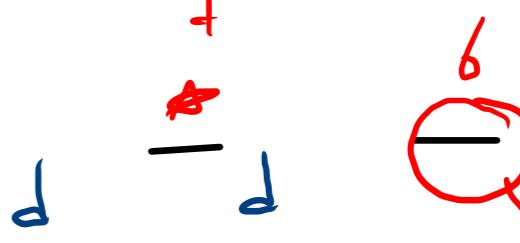
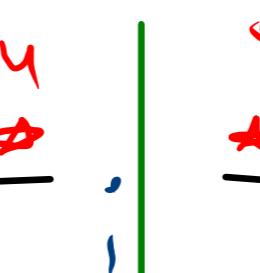
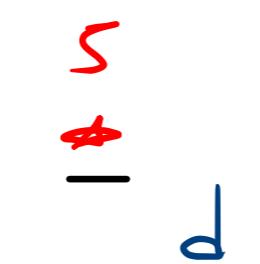
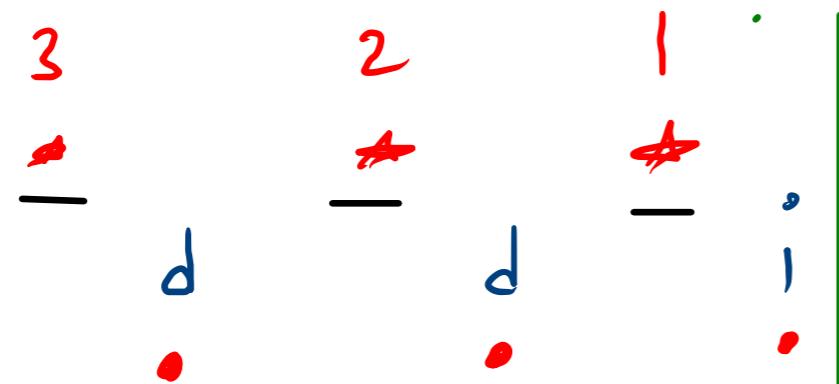
iidddd → 126543

1 2 3 4

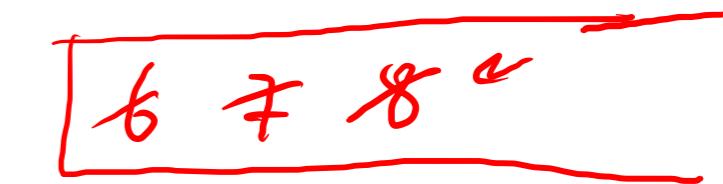
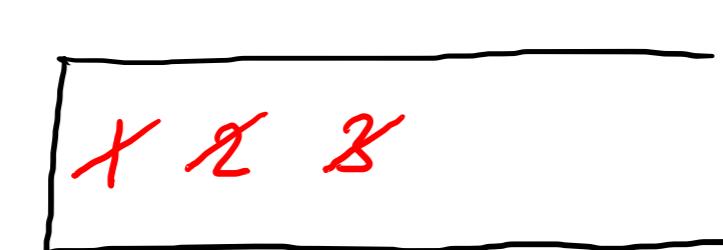
5 6 7 8



val  
1  
2  
3  
4  
5  
6  
7  
8  
9

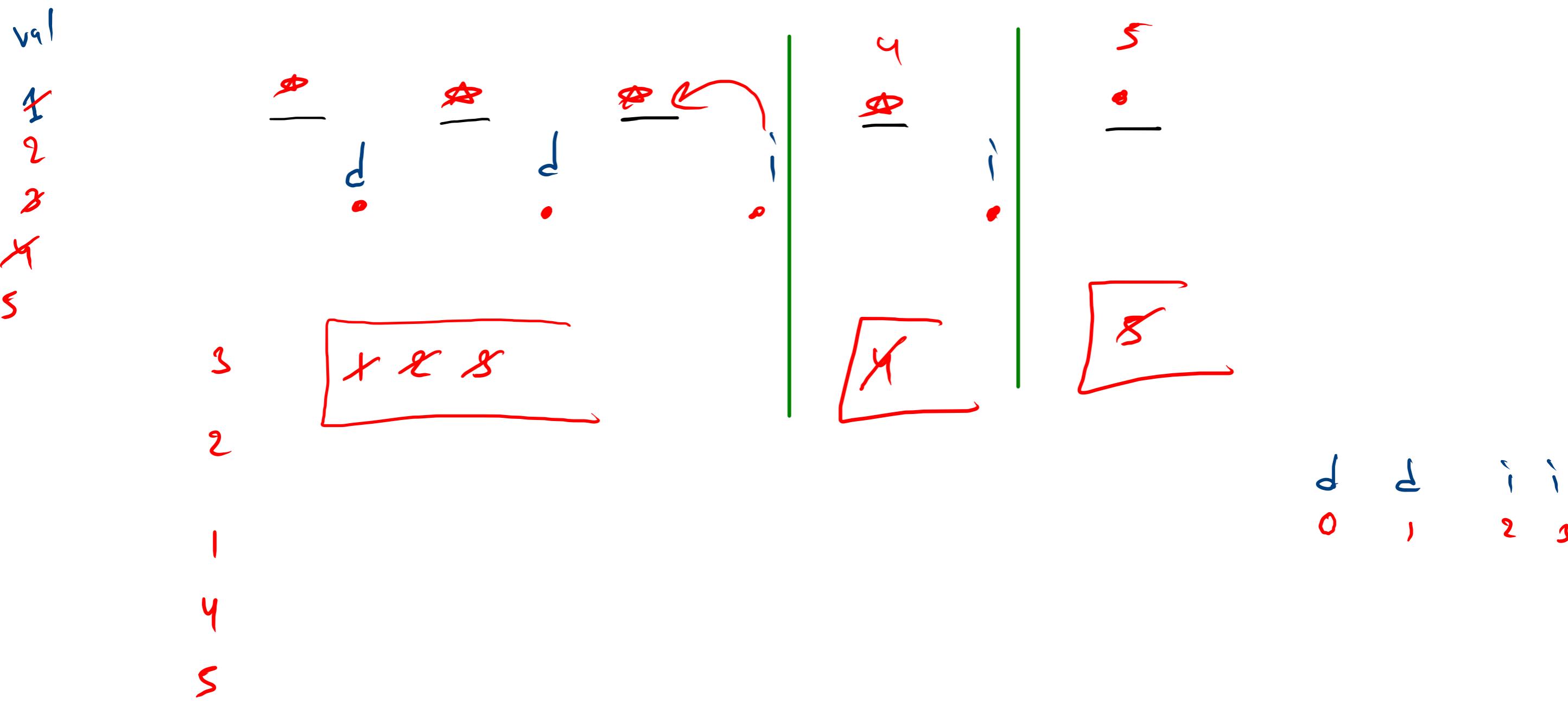


after loop



i add  
display all

d d i d i d  
0 1 2 3 4 5 6  
→

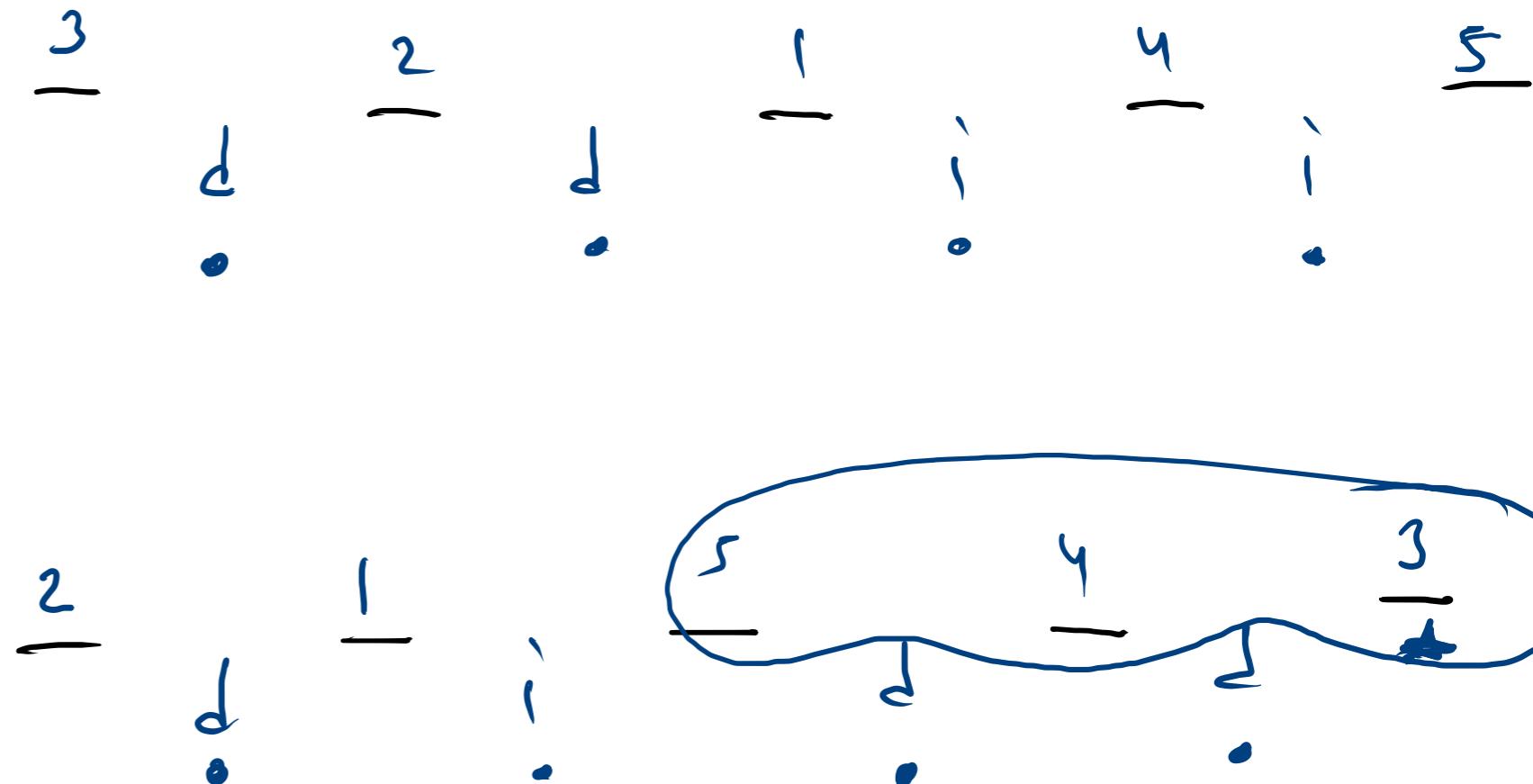


$\min x \otimes 45$

```
for(int i=0;i<str.length(); i++){
    char ch = str.charAt(i);

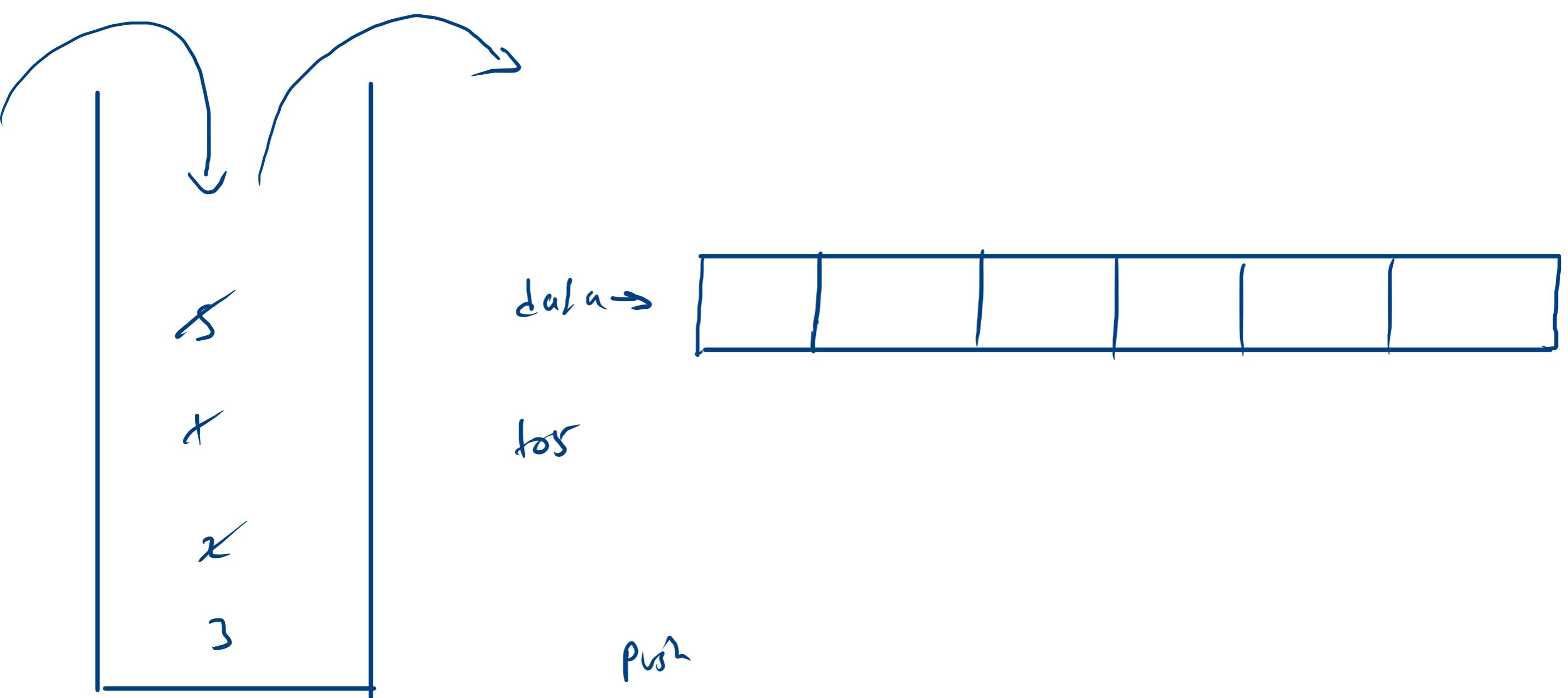
    if(ch == 'd'){
        st.push(min);
        min++;
    }else{
        st.push(min);
        min++;
        while(st.size() > 0){
            System.out.print(st.pop());
        }
    }
    st.push(min);
    while(st.size() > 0){
        System.out.print(st.pop());
    }
}
```

x x x x 8



x x x x 5

x x x 4 5



~~5 push 10 display~~

~~push 20 display~~

~~push 30 display~~

~~push 40 display~~

~~push 50 display~~

~~push 60 display~~

~~top pop quit~~

~~calc~~ c top

do loop  $\rightarrow$  bottom

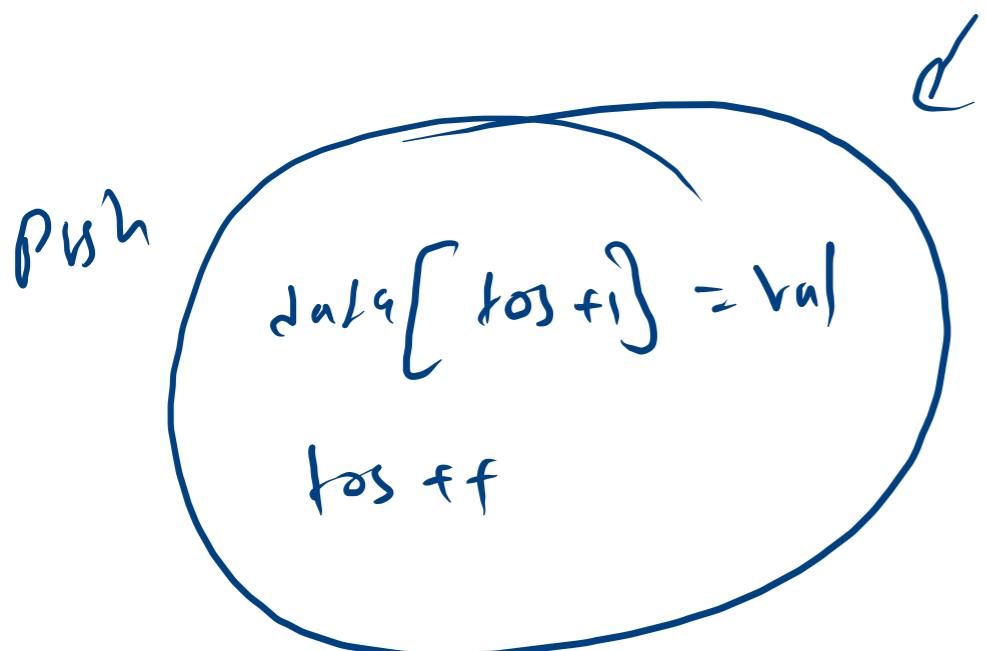
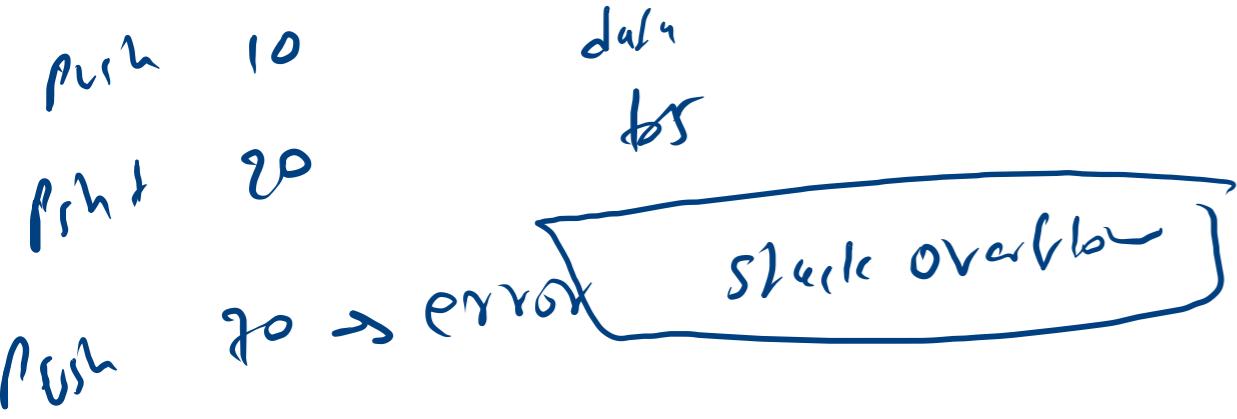
data

tos



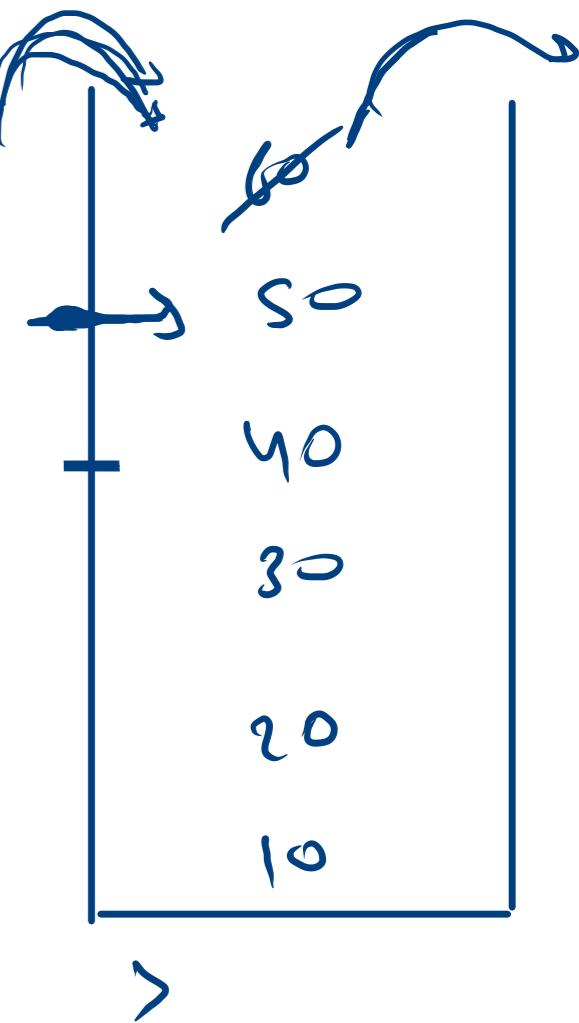
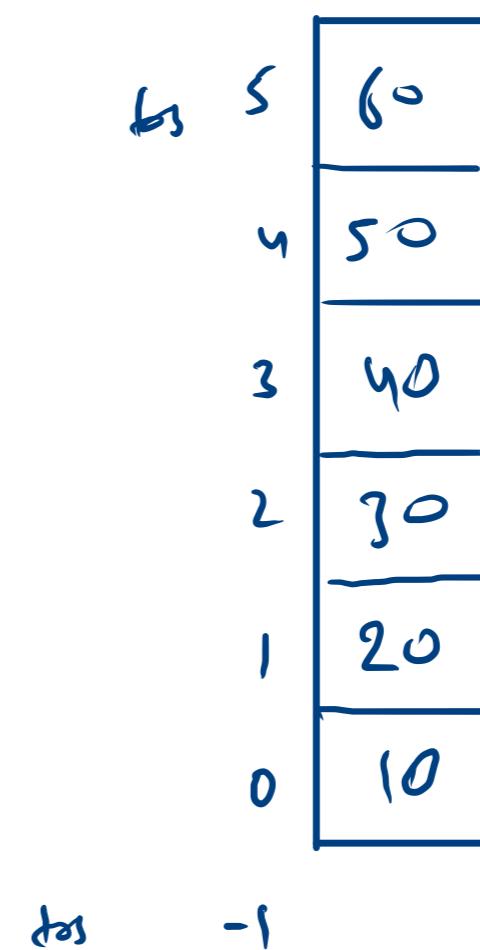
10			
20	10		
30	20	10	
40	30	20	10
60	60		

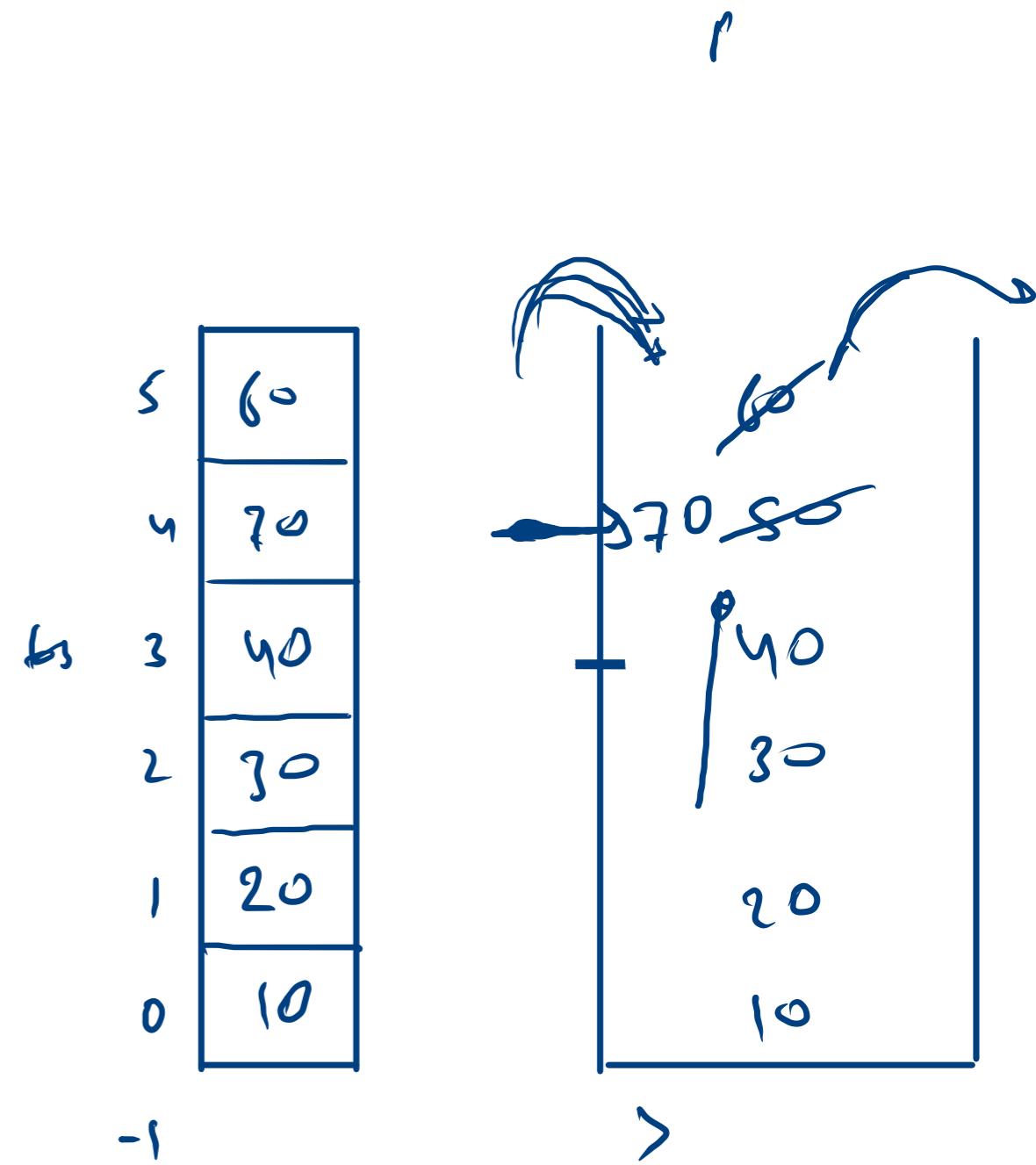
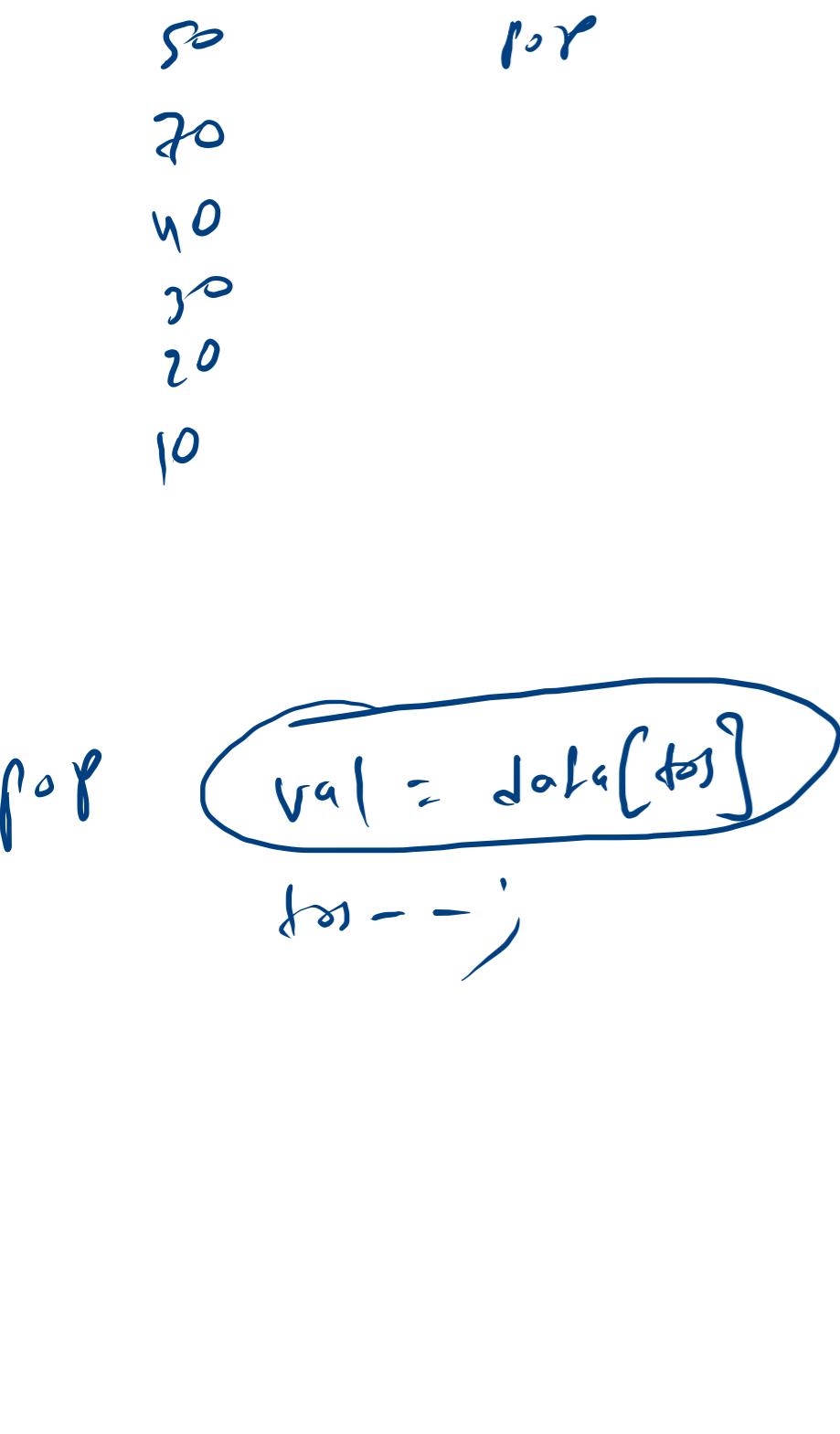
5
60
80
40
10
20
10

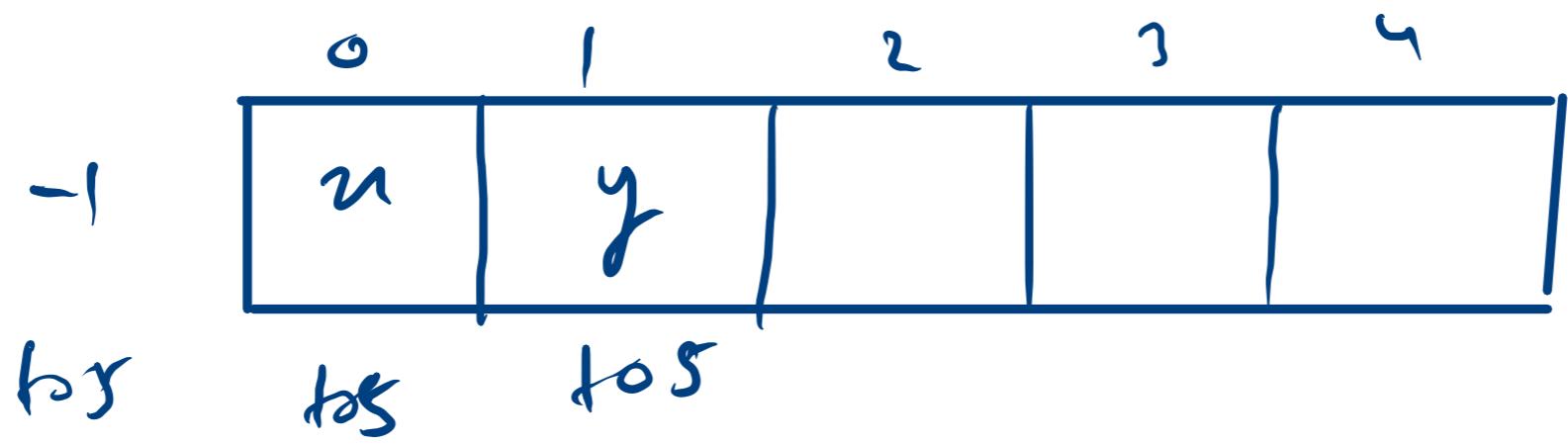


data[6] = 6  
 h = h - 1

pop

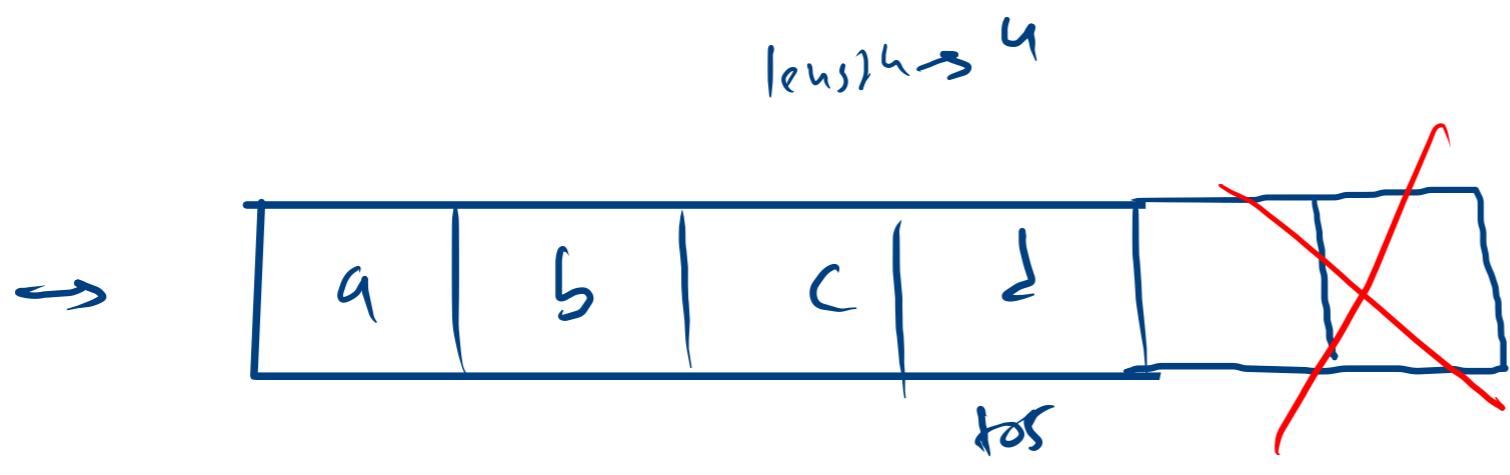




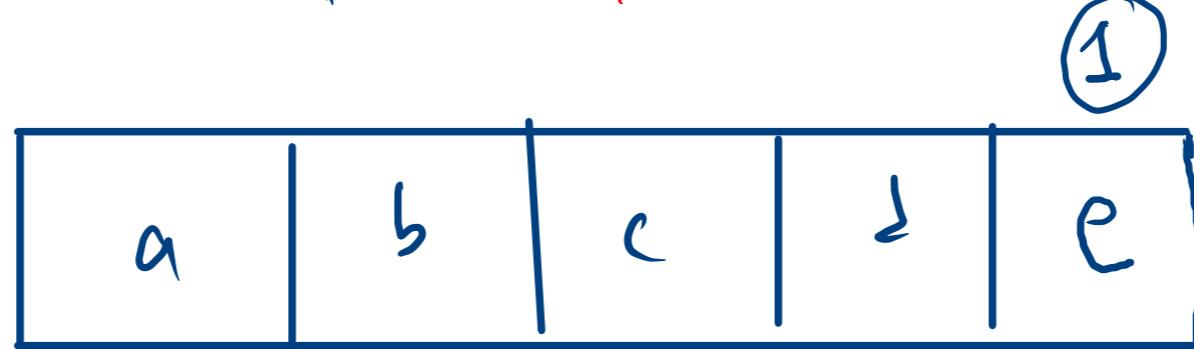


tos  
size

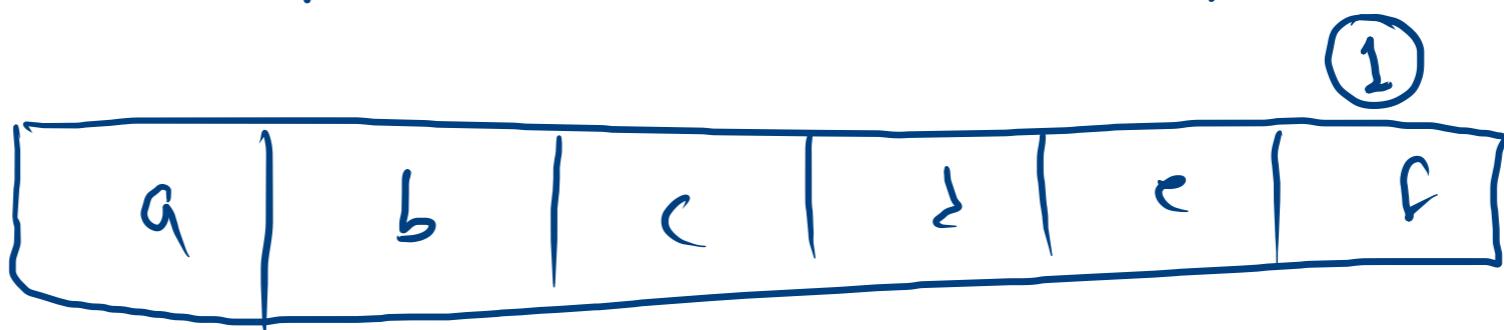
-1	0
0	1
1	2
tos	<u>tos+1</u>



push e



push f



[new arr  $O(1)$   
copy  $O(n)$   
size + 1  $O(1)$   
size +  $\infty$   $O(1)$

$$\frac{1000}{4} \leftarrow 100\text{y}$$

a	b	c	d
tos			

push e

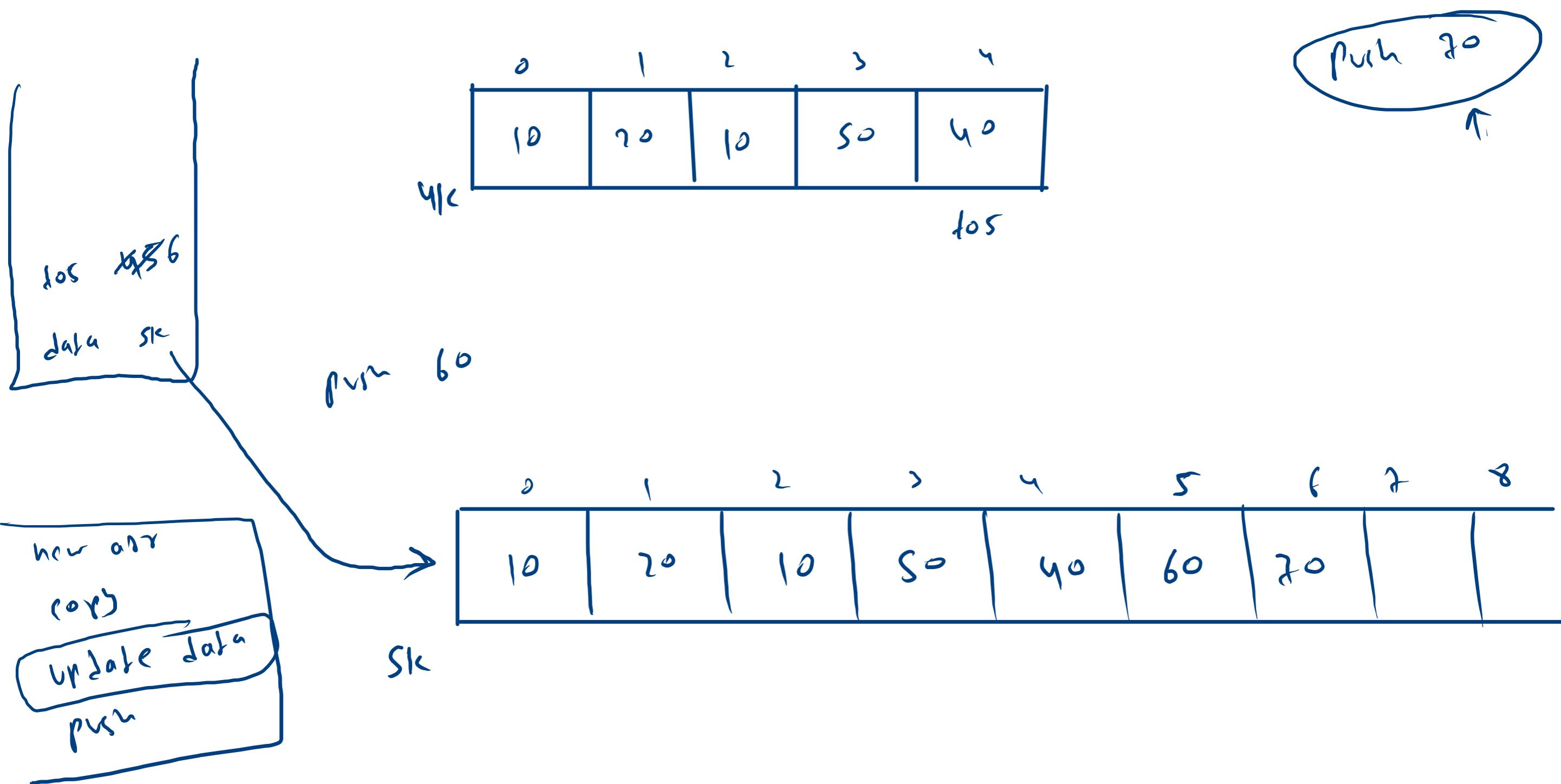
a	b	c	d	e	f	g	h
---	---	---	---	---	---	---	---

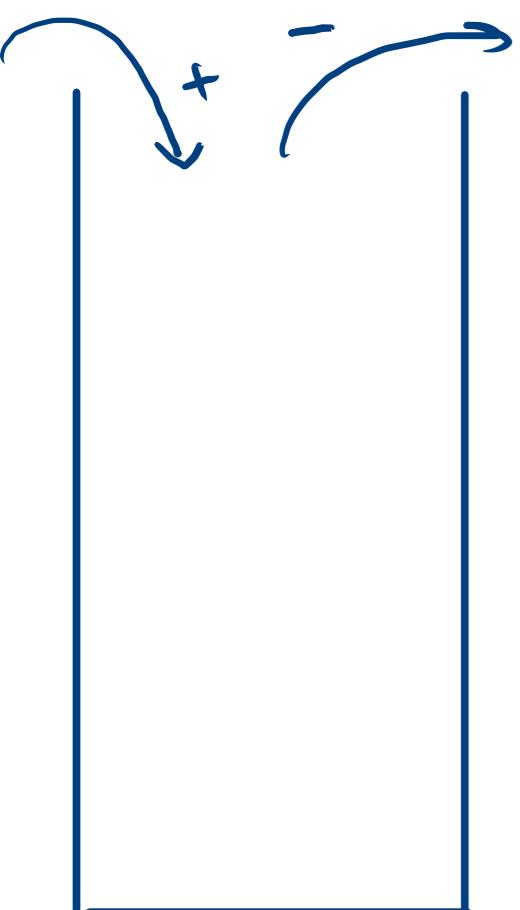


$\times 2$

$\rightarrow$  1.75

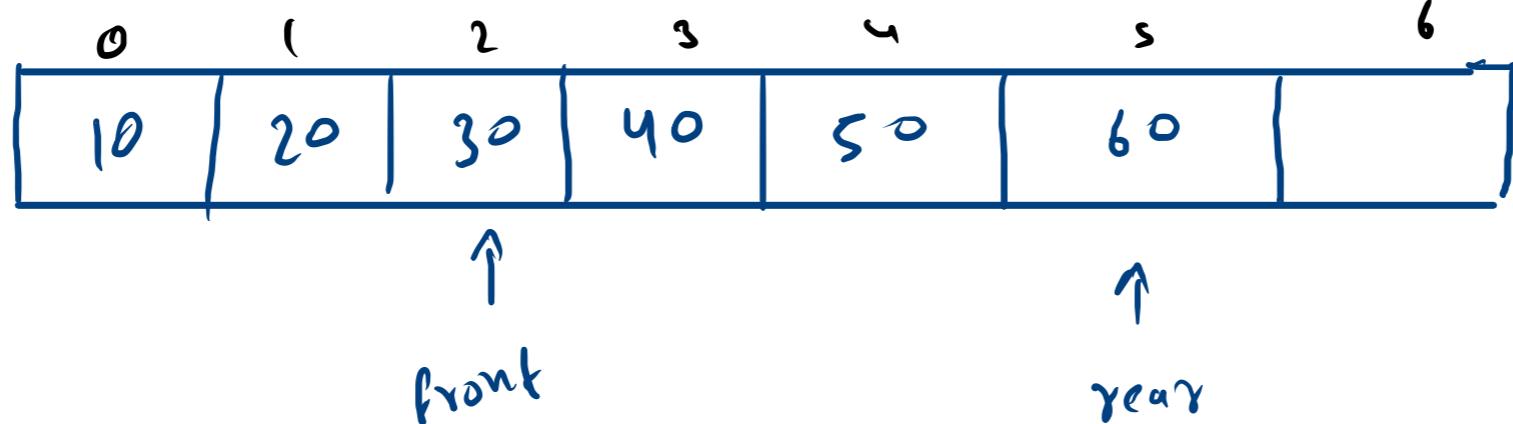
a	b	c	d	e	f	s	h	i	j	k	l	m	n	p	p
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---





last

data →



$$\text{size} = 48484$$

$$\text{remove} \rightarrow 10$$

$$\text{add} \rightarrow 60$$

$$\text{remove} \rightarrow 20$$

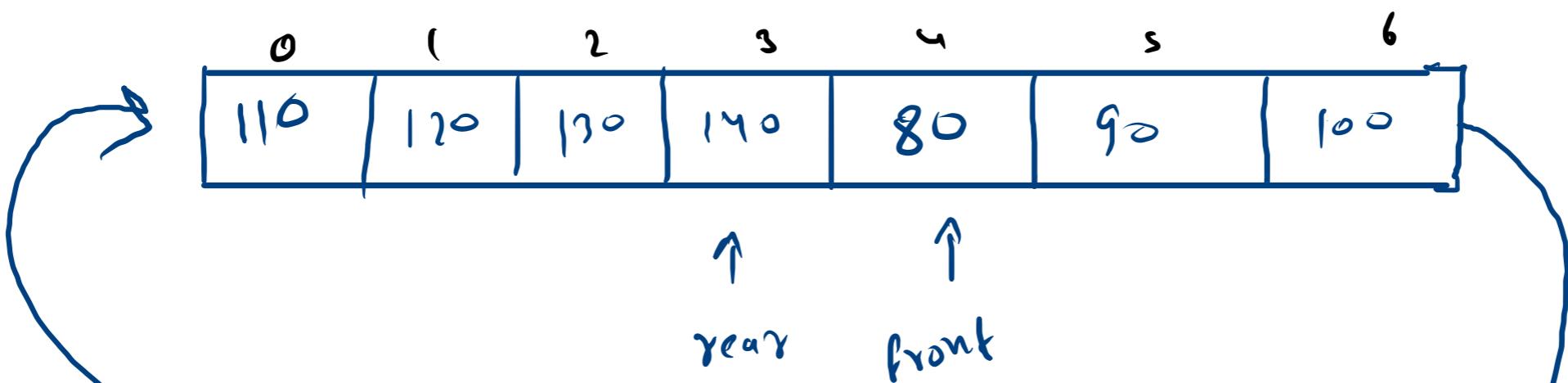
$$\text{year} = \text{front} + \text{size} - 1$$

$$\text{to be added} = \text{front} + \text{size}$$

year  
front

$y_0 s_0 \leftarrow$

$O(n)$

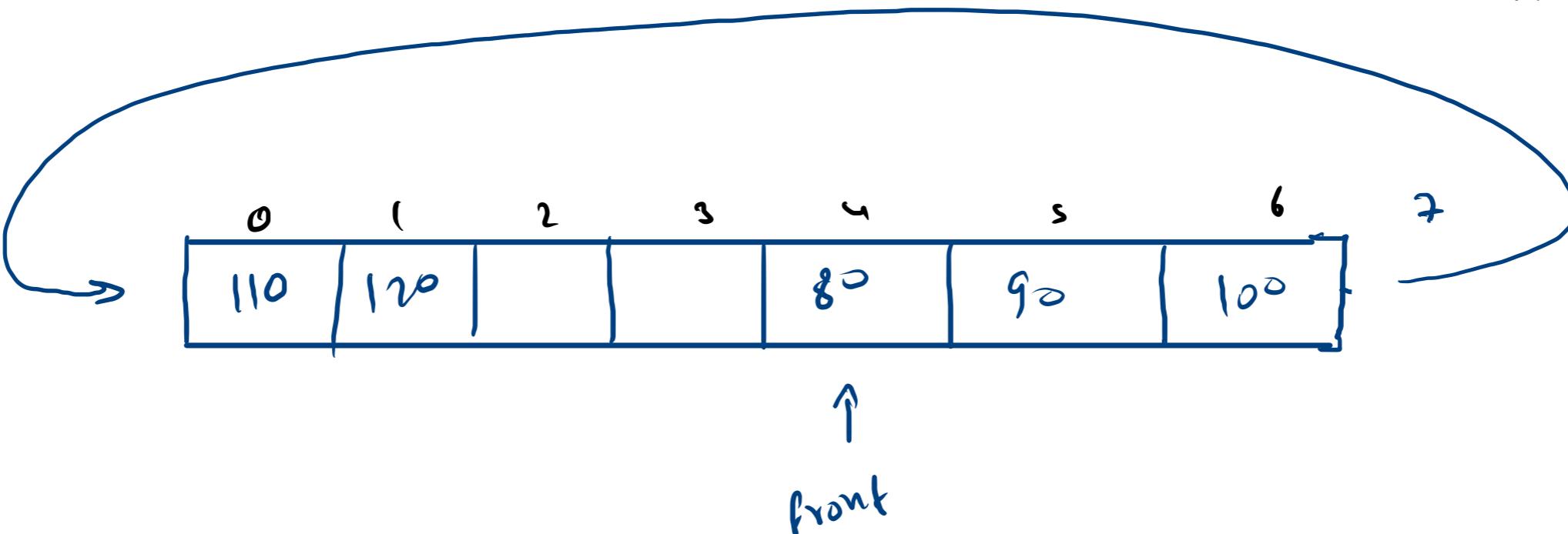


$\text{size} = 7$

to be added

add 70  
year  $\rightarrow$  30  
add 80  
year  $\rightarrow$  40  
add  $\rightarrow$  90  
add 110

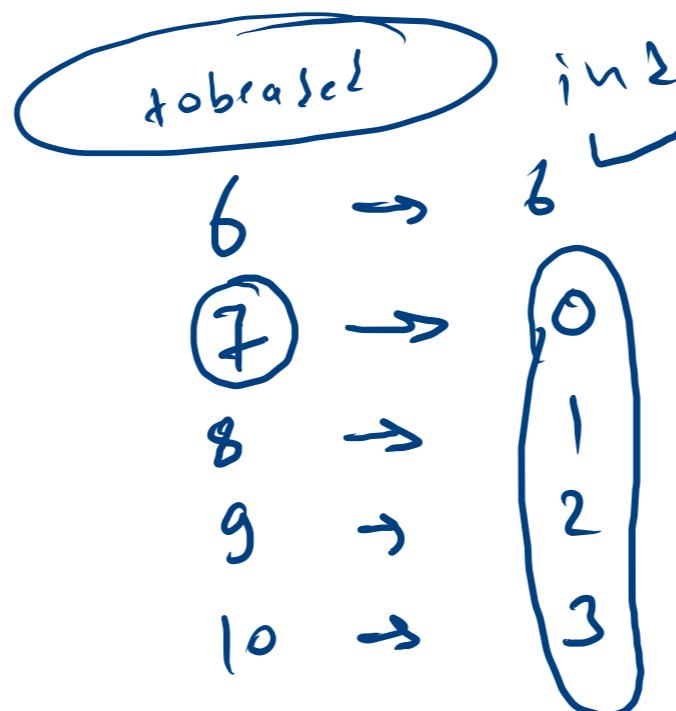
$h = 2$



size

$\approx 8^4$

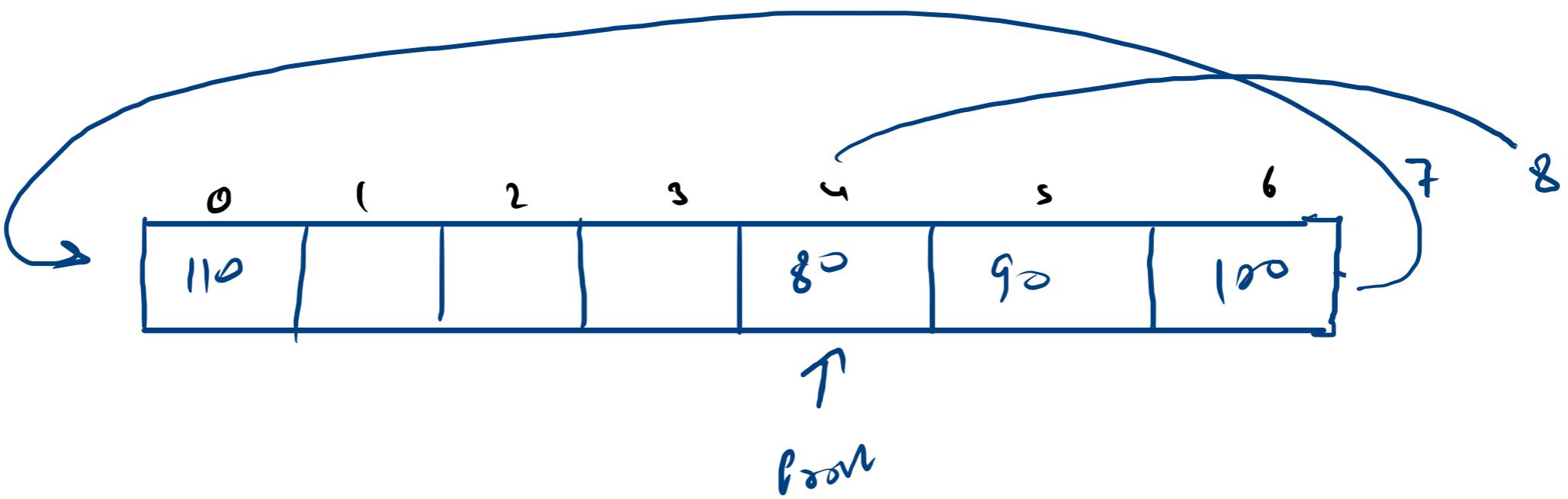
to be added  $\rightarrow$  front + size



$$in2 = to be added - 1$$

add 10  
110

$h = 2$



size  $\times 8^4$

add  $\rightarrow$  front+size

6  
6

7  $\rightarrow$  0  
8  $\rightarrow$  1

add  $\rightarrow$  add - n

$h = 6$

```

void add(int val) {
    if(size == data.length){
        System.out.println("Queue overflow");
    }else{
        int ind = front+size;
        if(ind >= data.length){
            ind = ind-data.length;
        }
        data[ind] = val;
        size++;
    }
}

```

0	1	2	3	4	5	6	7
50	60	10	20	30	40		

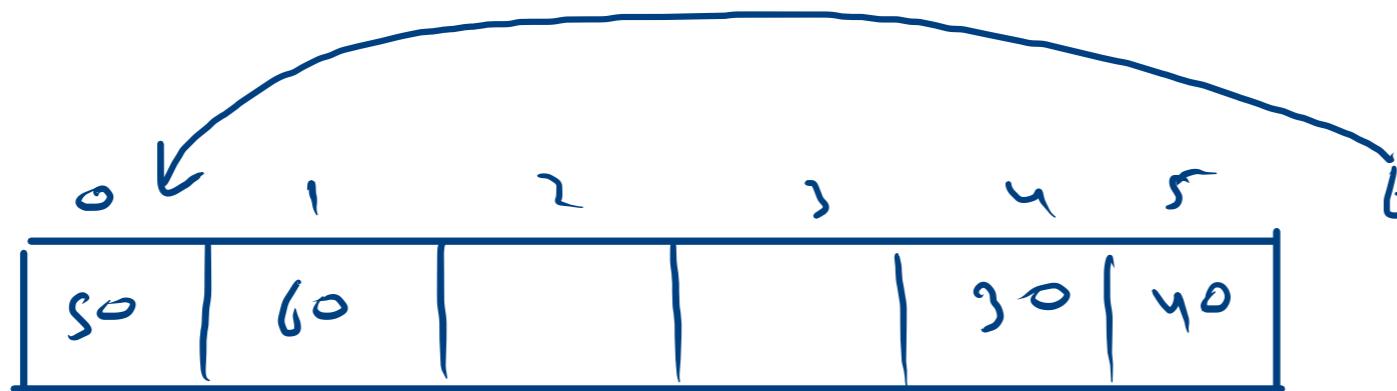
front

size  $\rightarrow$  2  $\times$  2  $\neq$  6

in2  $2+2 = 4$   
 in2  $2+3 = 5$   
 in2  $2+4 = 6$   
 in2  $2+5 = 7$

6  $\rightarrow$  0      in2-h

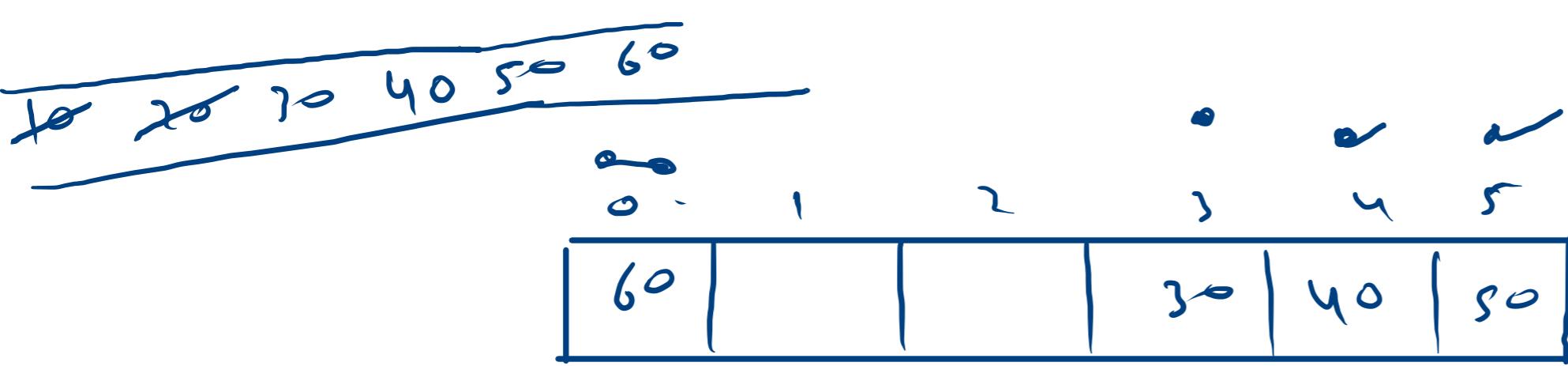
7  $\rightarrow$  7-6 = 1



from

size 4

remove 30  
remove 40



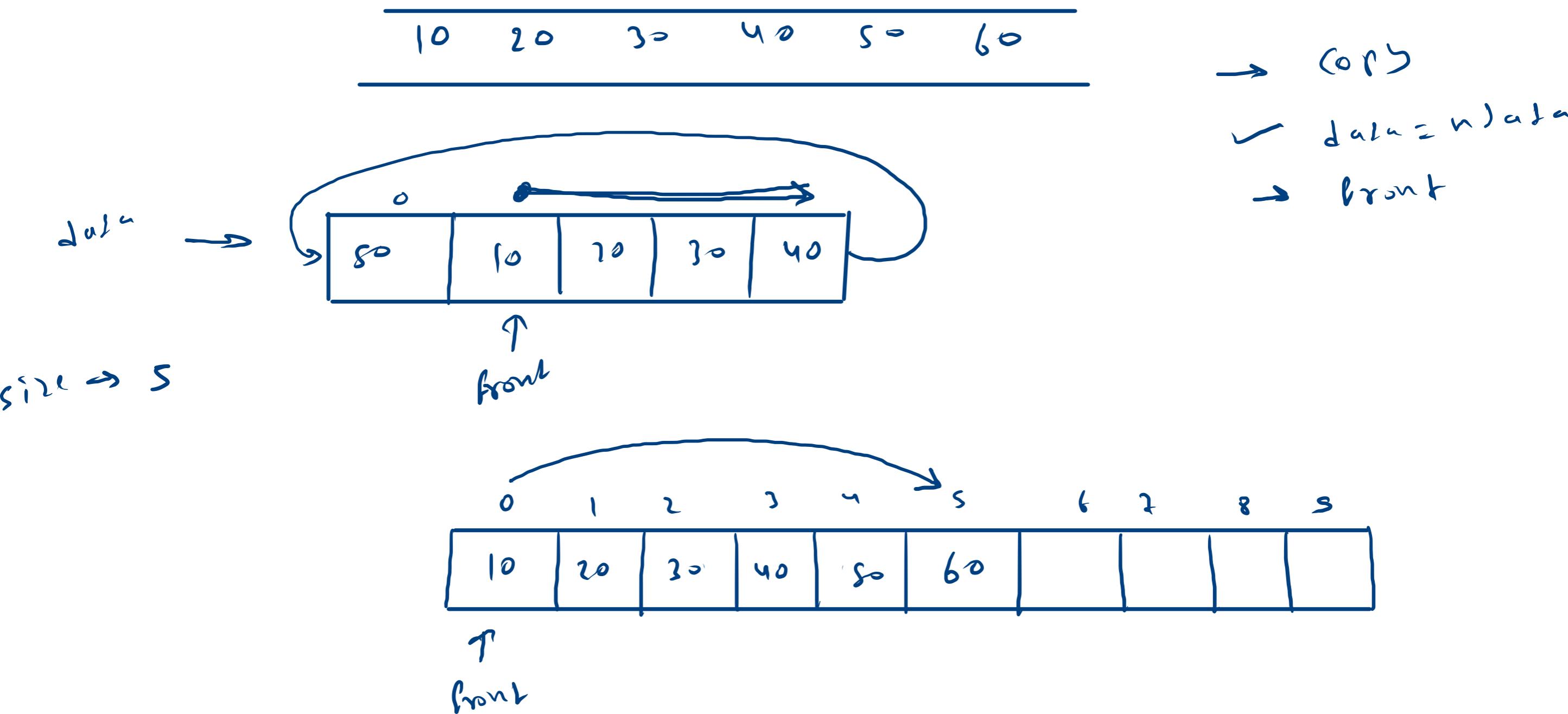
$T_{front}$

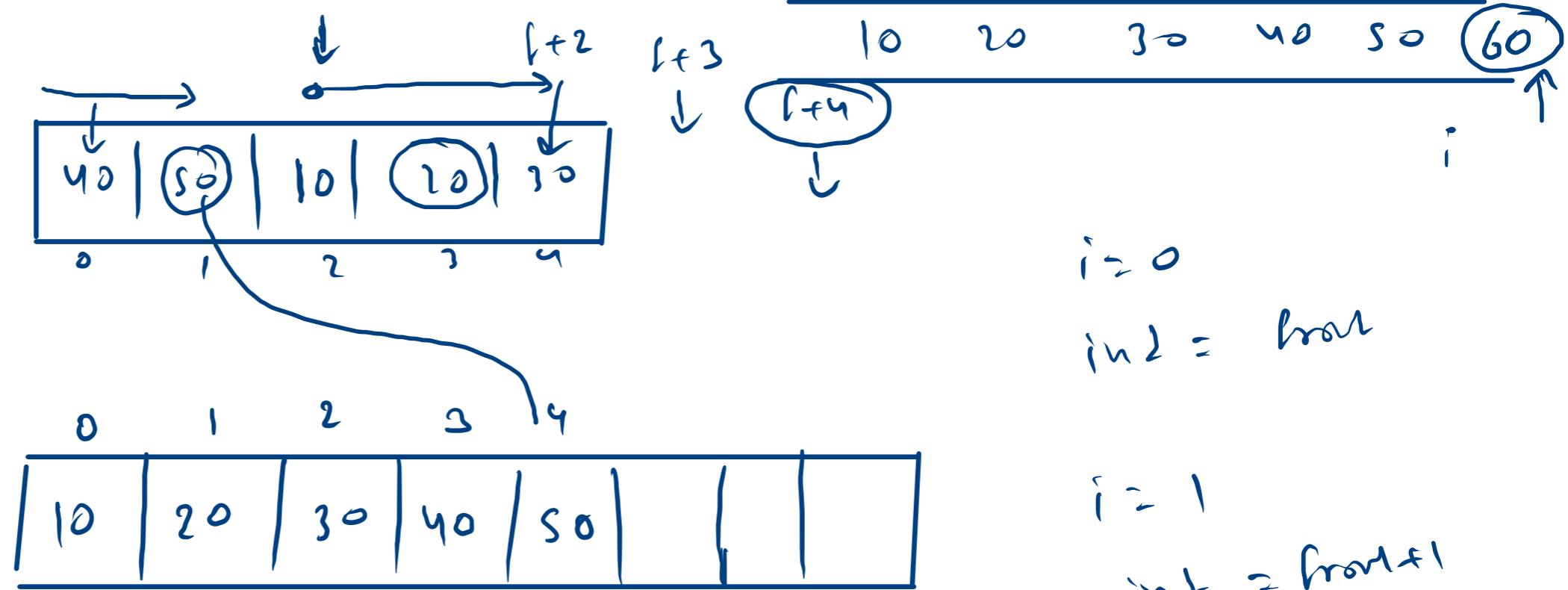
$\sin \approx 4$

0 - - 3

$152 \rightarrow front + 0$   
 $\rightarrow front + 1$   
 $front + 2$   
 $\vdots$   
 $front + 3$

10 20 30





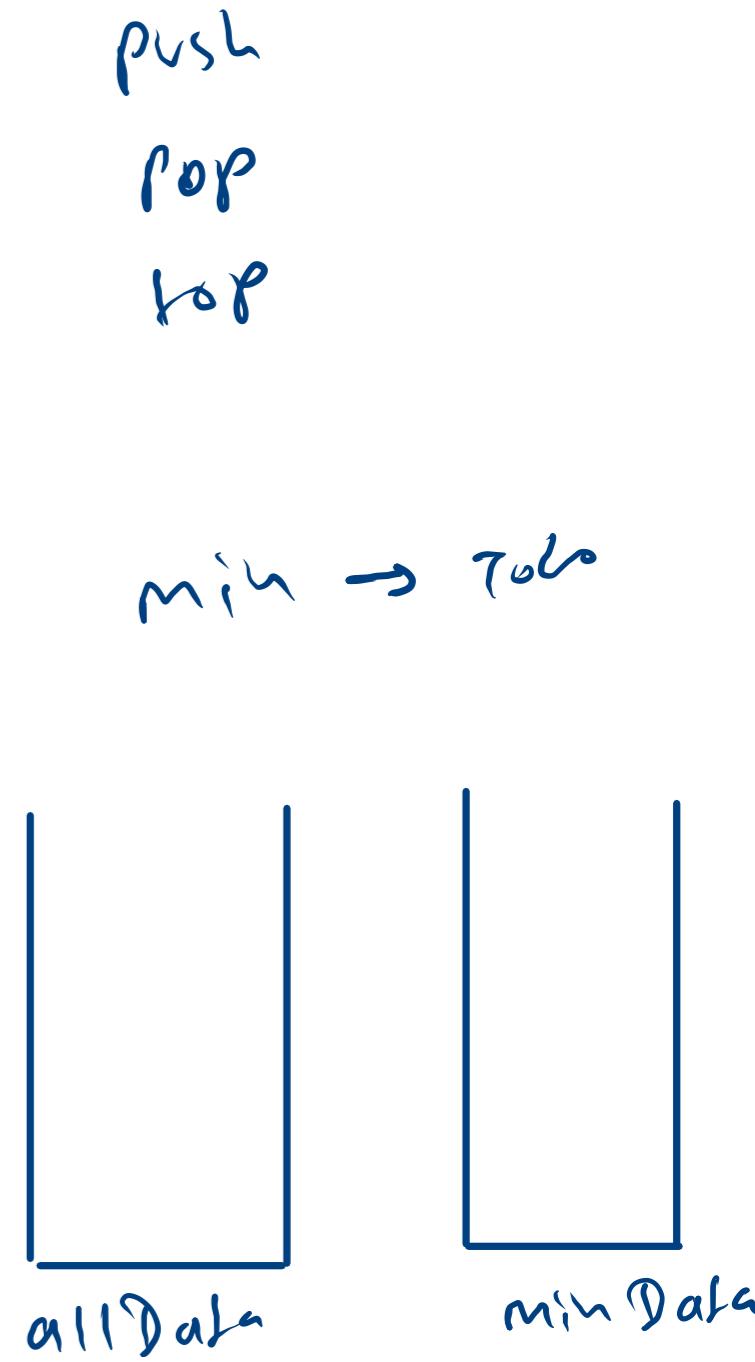
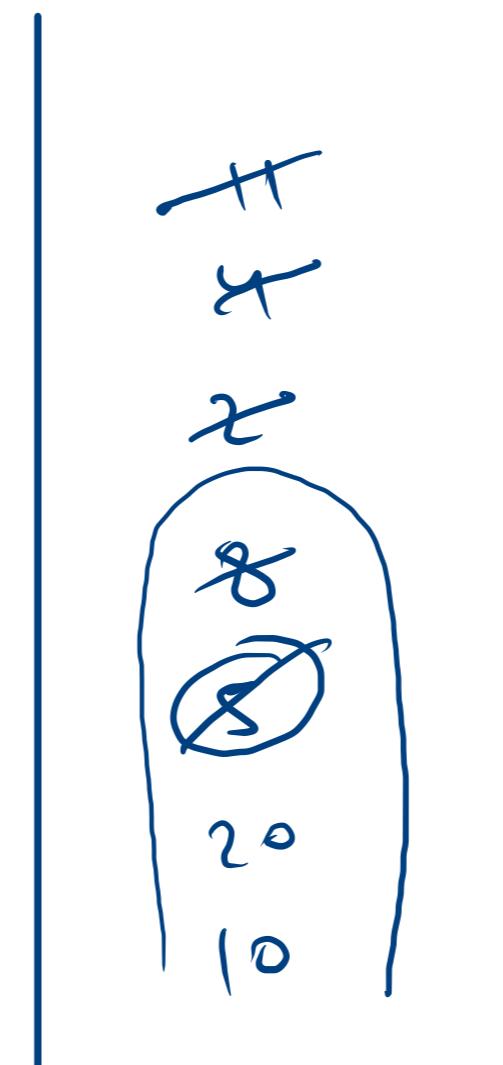
```

int ndata[] = new int[data.length*2];
for(int i=0;i<size;i++){
    int ind = front+i;
    if(ind>=data.length)ind = ind-data.length;
    ndata[i] = data[ind];
}
data = ndata;
front = 0;

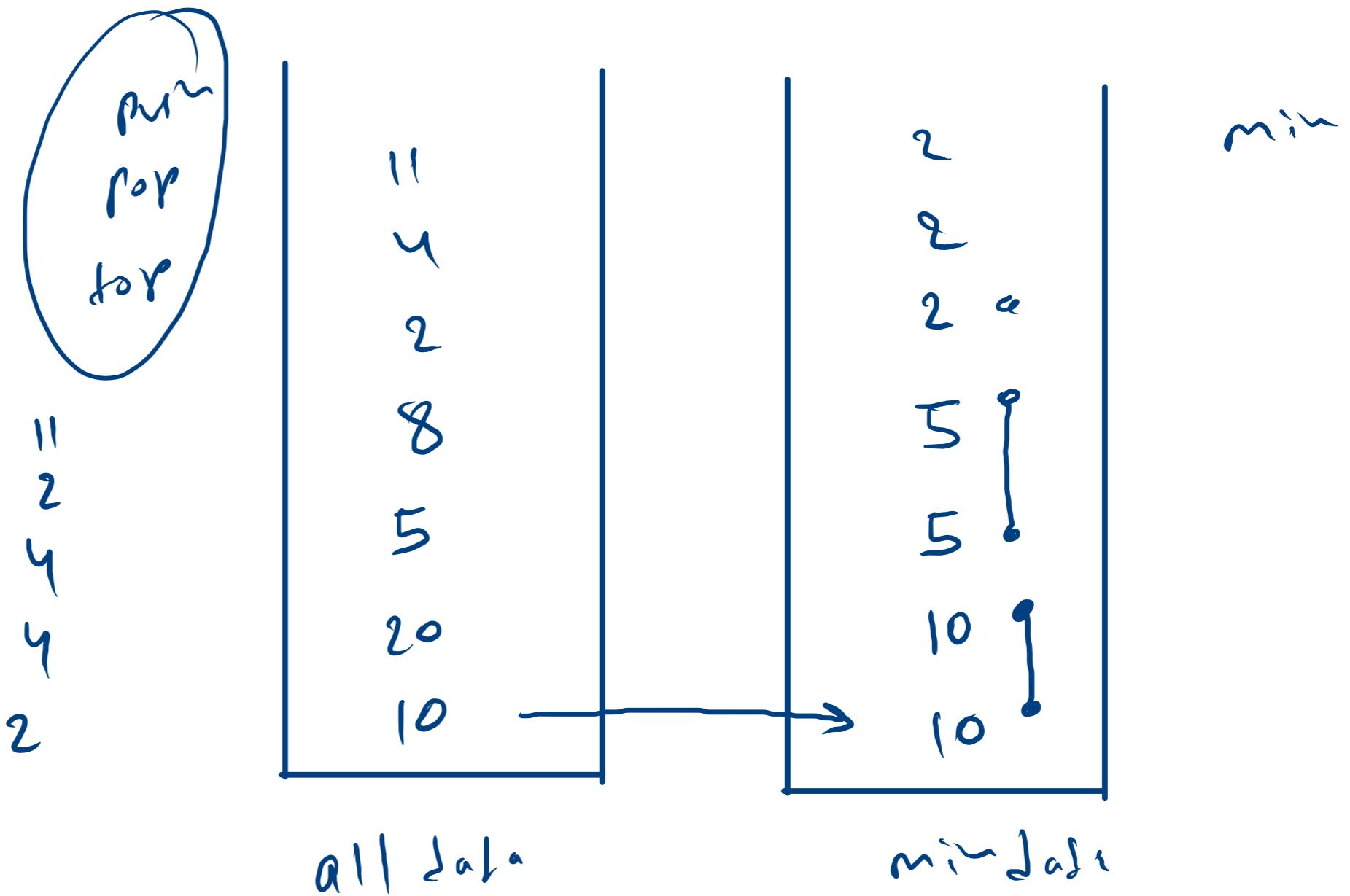
```

~~push 10 push 20~~  
~~push 5 push 8~~  
~~push 2 push 4~~  
~~push 11 top min~~  
~~pop top min pop~~  
~~top min pop top~~  
~~min pop top min~~  
~~pop top min pop~~  
top min pop quit

11  
2 ← min  
11  
4  
2 ← min  
4  
2  
2 ← min  
2  
8  
5 ← min  
8  
5  
5  
5



~~push 10 push 20~~  
~~push 5 push 8~~  
~~push 2 push 4~~  
push 11 ~~top~~ min  
pop top min ~~pop~~  
top ~~min~~ pop top  
min pop top min  
pop top min pop  
top min pop quit



108

