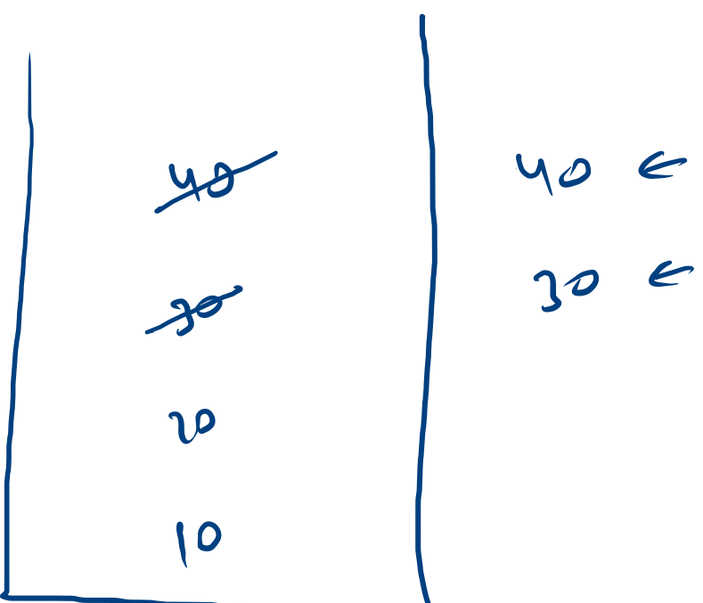
Stack

LIFO



Queue

FIFO

10
20
30
40

add

remove

get

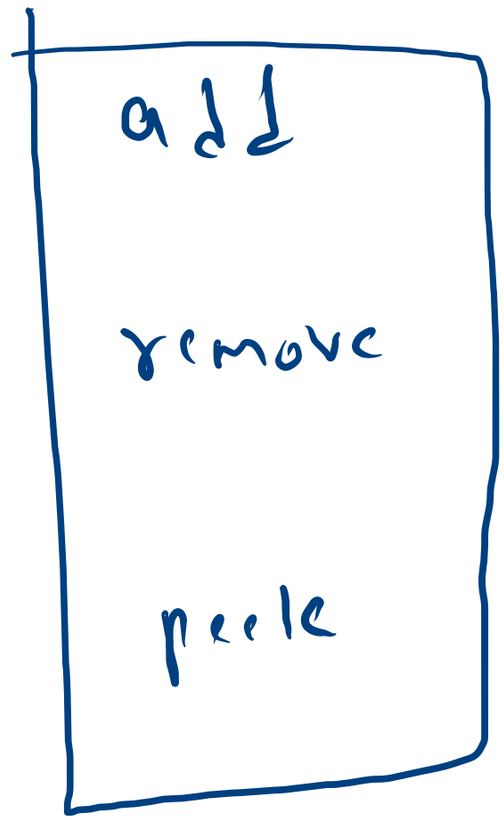
stack

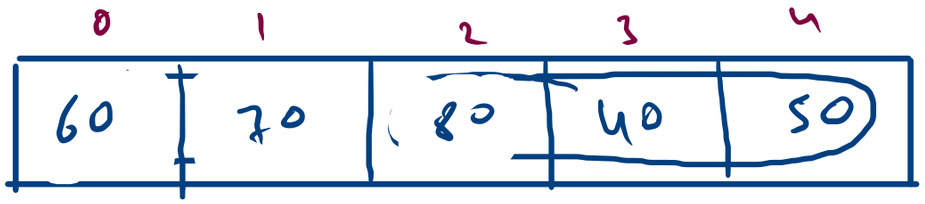
push

pop

peek

queue

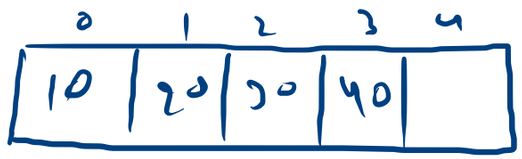




data →

front

Rear
index



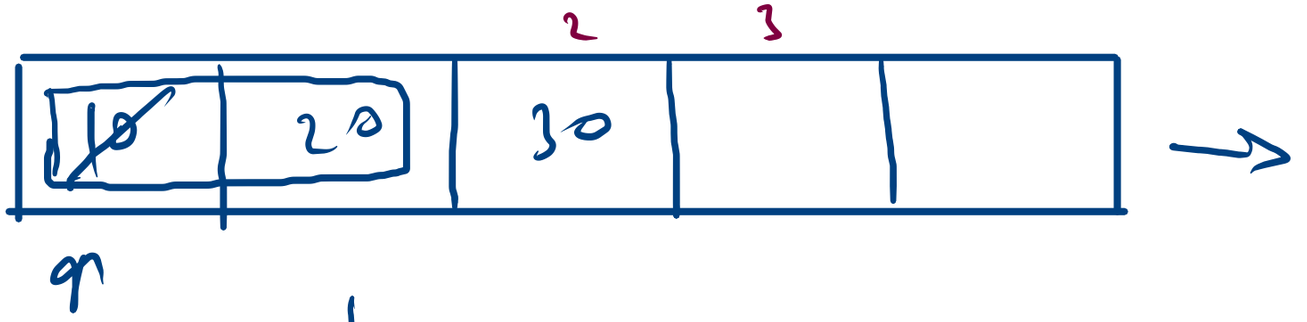
front

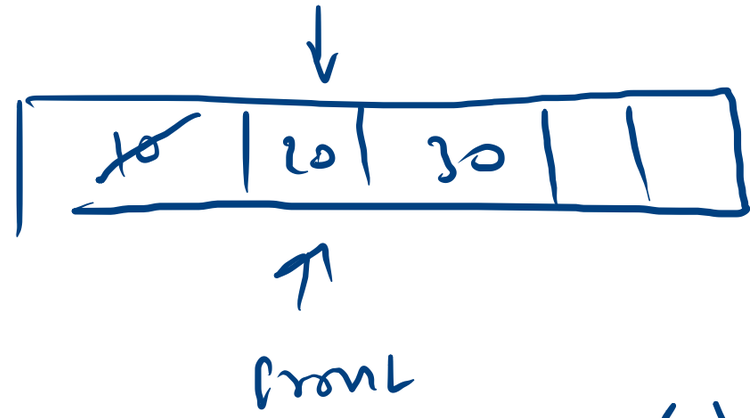
size 0 1 2 3

(front + size) to be add

size 1 2 3 4 5 4 3 2 1 0

add 10
add 20
add 30
remove 10
add 40
display 20 30 40
remove 30
add 50
add 60
display 20 30 40 50 60



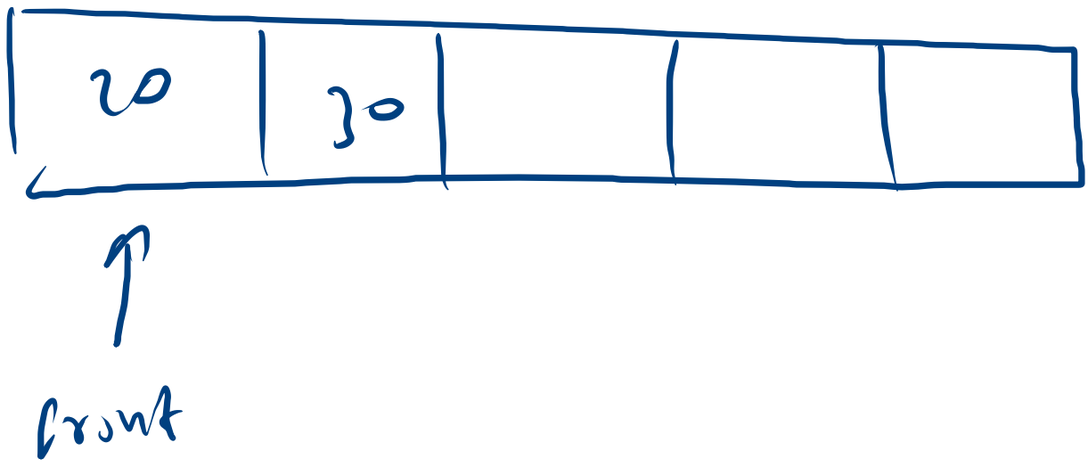


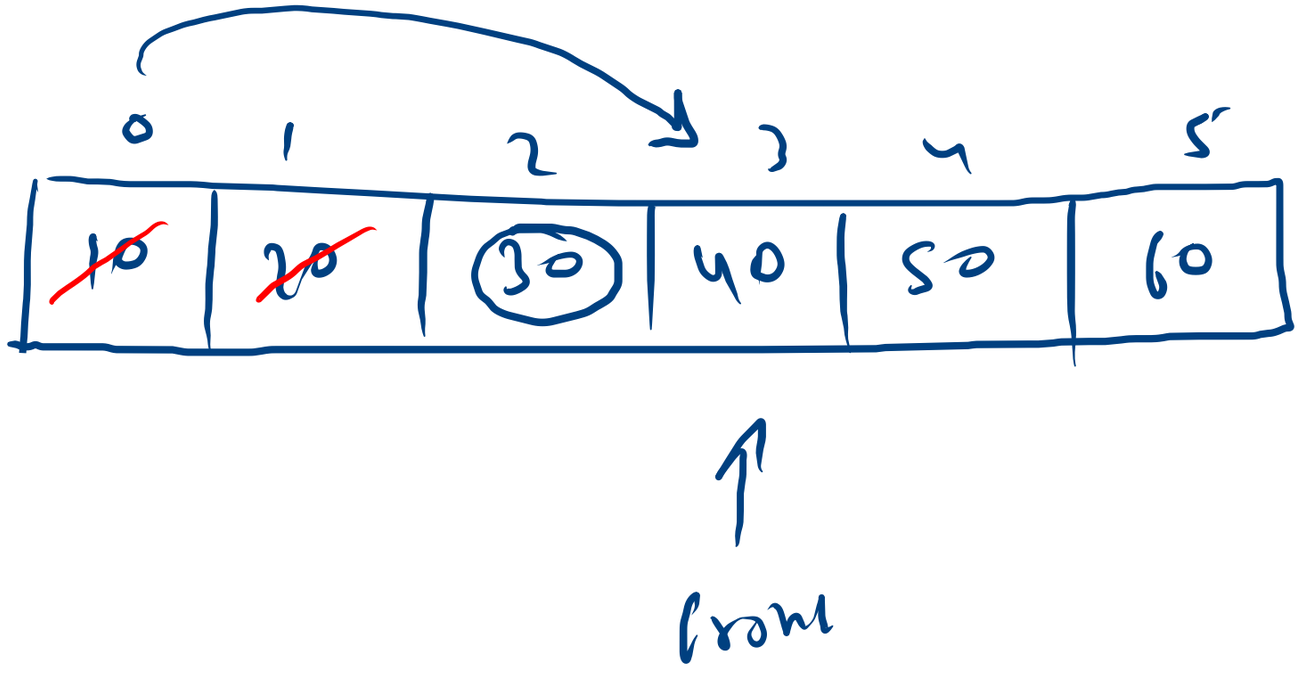
size 3

O(n)

size 2

O(1)

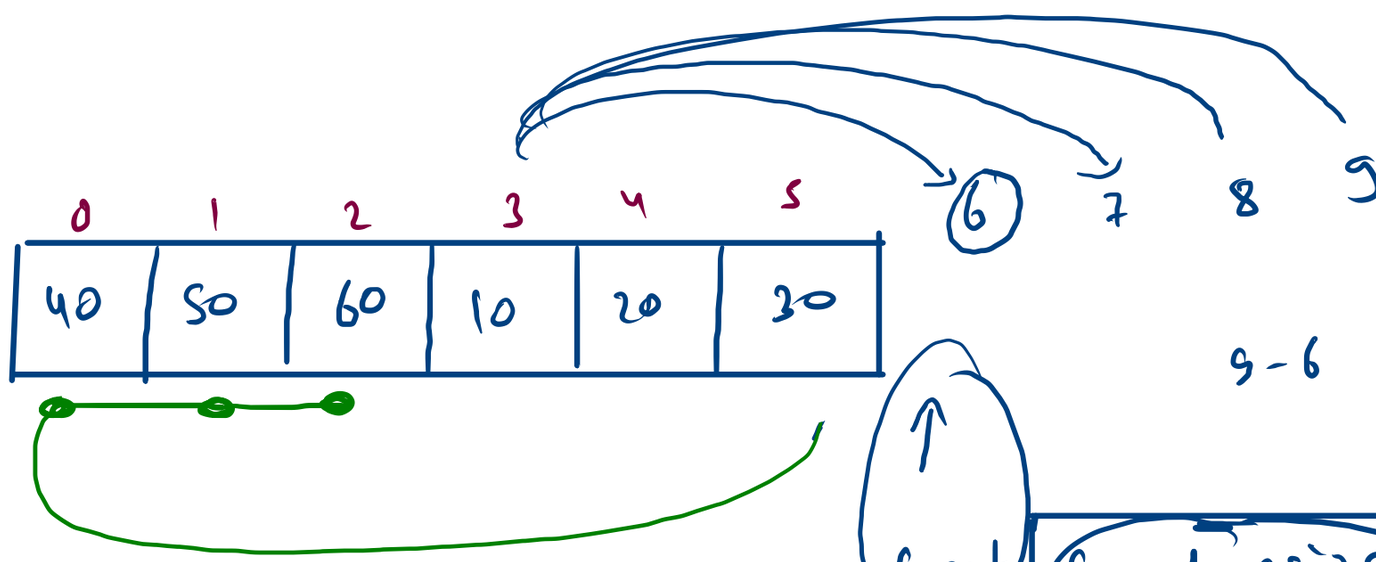




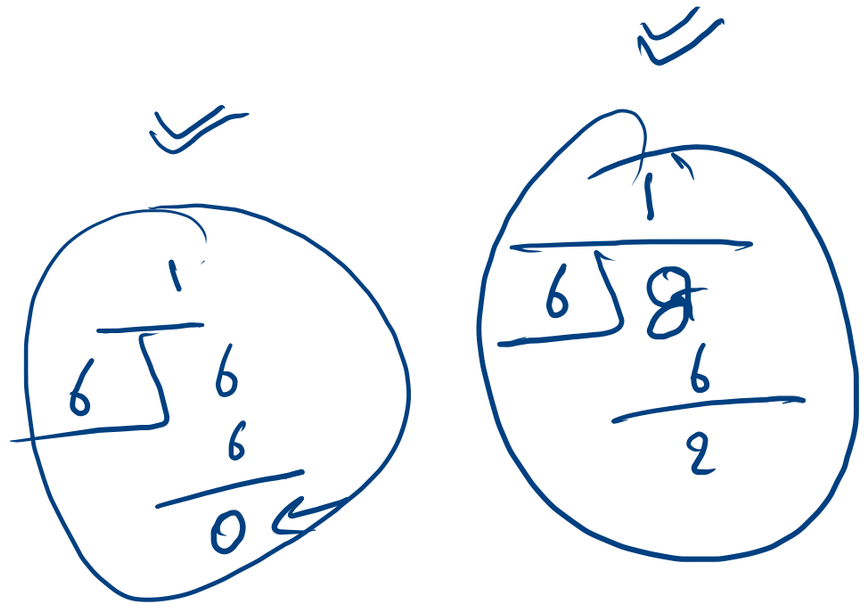
all 60

size 0 1 2 3 4 5 6

To add = front + size



index -



size 8 4 8 6 8
4
3

-h

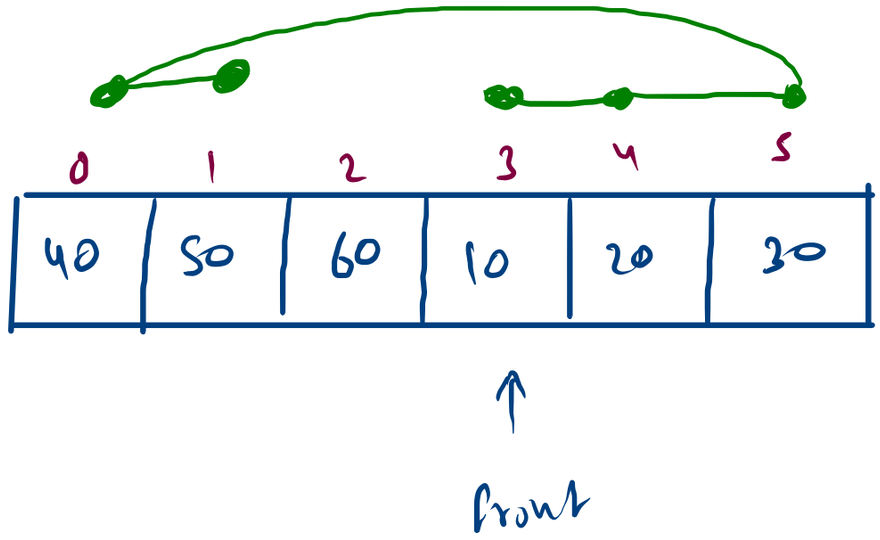
9-6

n = 6

32 32 32 idx of h

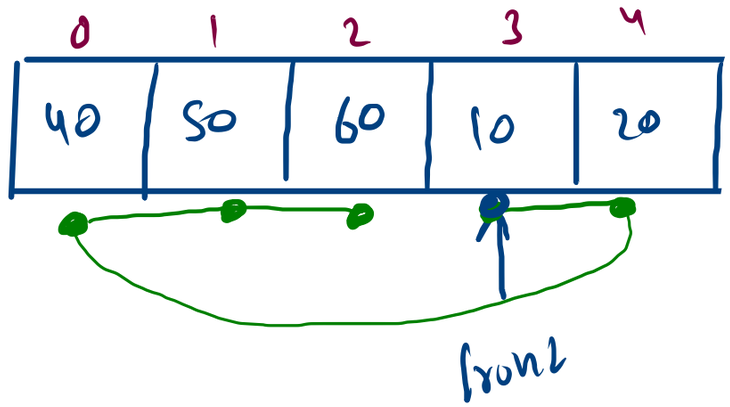
front	fsize	-h	idx of h
6	0	0	0
7	1	1	1
8	2	2	2

Abstraction

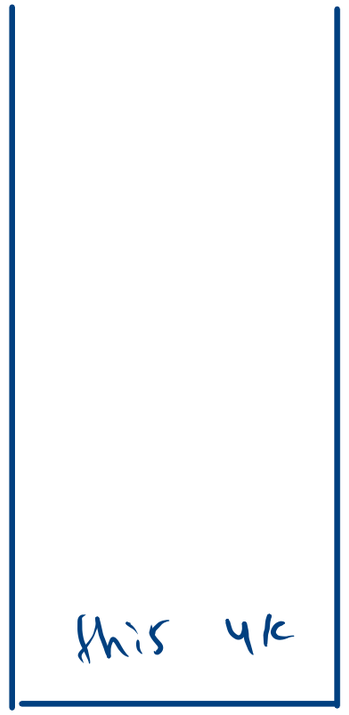


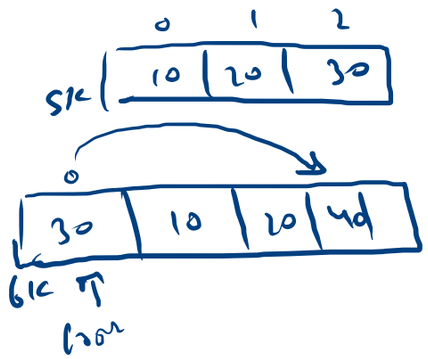
	0	1	2	3	4
front+i	3+0	3+1	3+2	3+3	3+4
	3	4	5	6	7

size = ~~6~~ 5

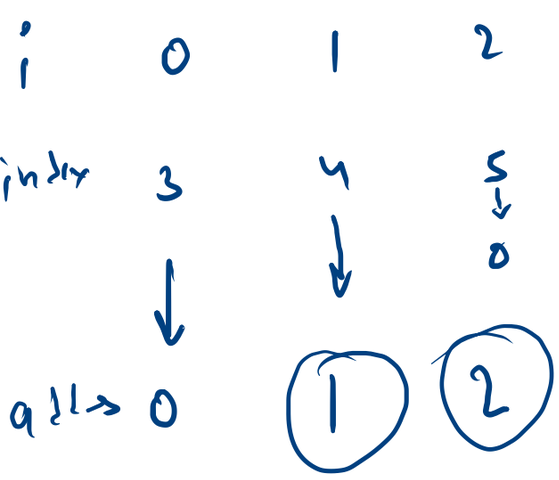


size 5

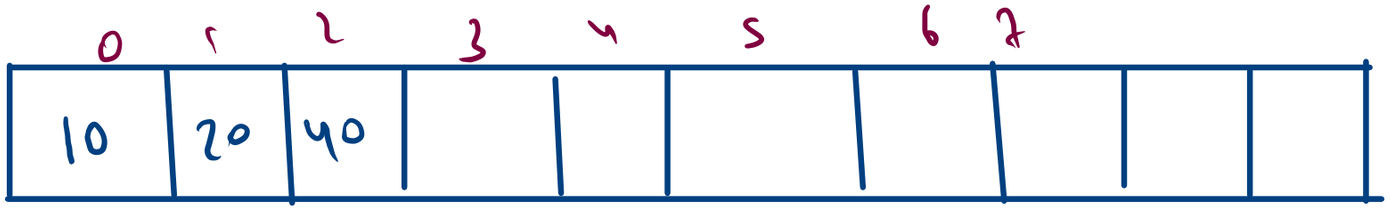




data	~~5k~~ 6k
front	0
size	3



front size



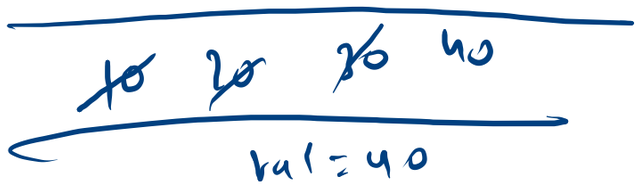
main d

val = 40 val = 30

push
pop
size
top

helper d

add
remove
peek



val = 40

push ellicient → O(1)

pop

pop 40

Top

```
while(mainQ.size() > 1){
    int val = mainQ.remove();
    helperQ.add(val);
}
int val = mainQ.remove();
helperQ.add(val);
while(helperQ.size() > 0){
    mainQ.add(helperQ.remove());
}
return val;
```

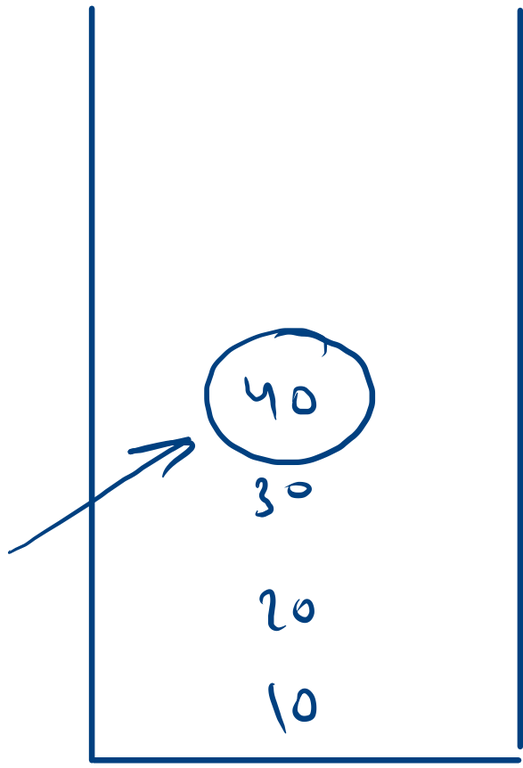
m

10	20	30	40	10	20	30	40

val = 40

h

10	20	30	40



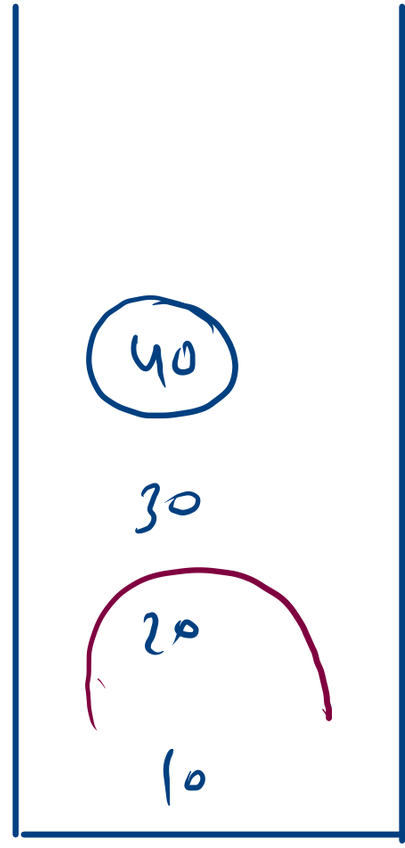
main2

30 | 20 | 10 | 40 |

40 | 30 | 20 | 10 |

pop

~~10~~ | ~~20~~ | ~~10~~ | 30 | 20 | 10 |



main Q

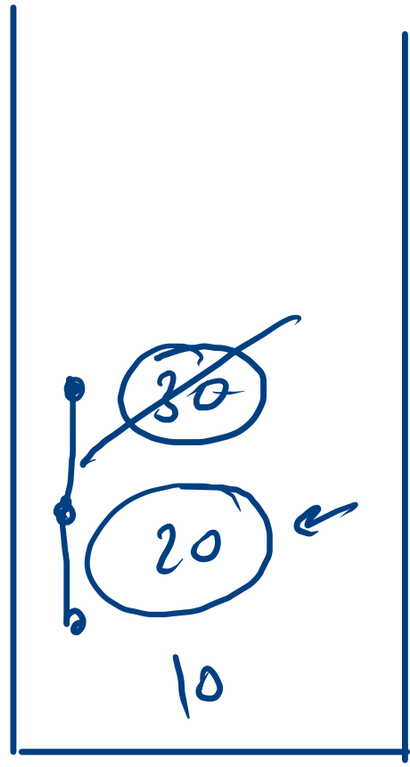
10	20	10	30	20	10

2
3

pop

30	20	10	40	30	20	10	50

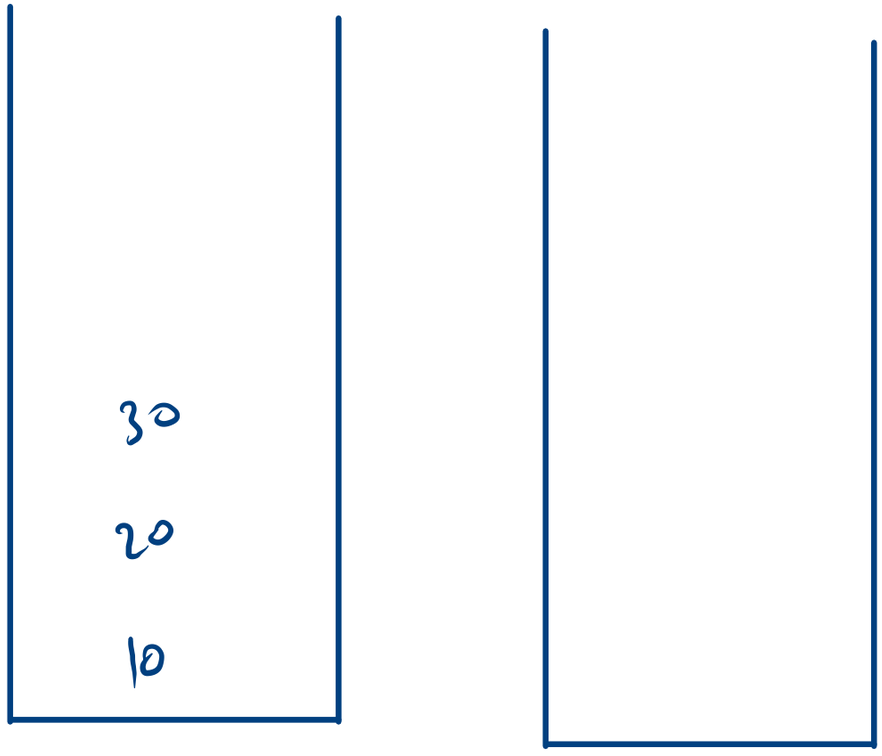
size 4



add O(1)
remove O(n)

add O(n)
remove O(1)

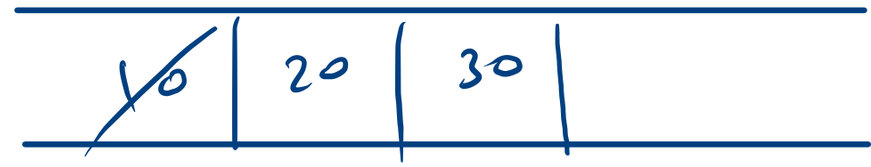
add O(1)
remove O(1)



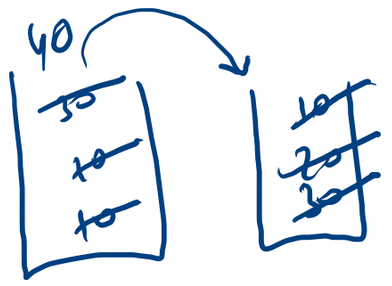
main s

helps

add efficient



O(1)



90
80
70
60
50
40
30
20
10

10
20
30
10
20
10

20
10
10

pop O(1)
add O(n)

10 | 20 | 30

helper

10 20 30 40 50 60 70 80

90

20
10

70
80
60
10
20
30

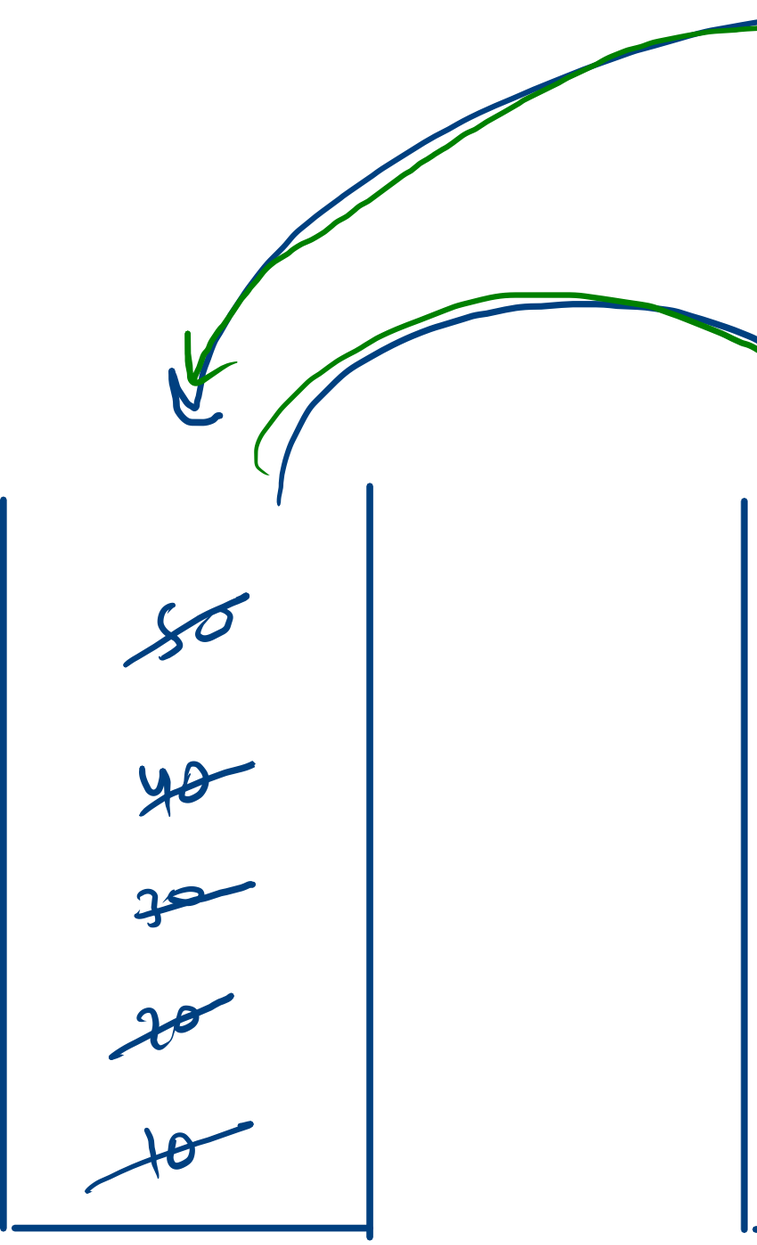
add

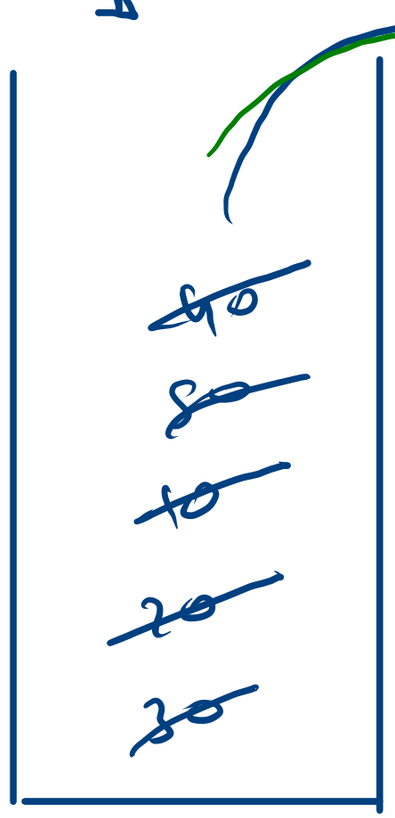
remove

mains

helper is empty
helper ← main

helper.pop





	10	20	30	40	50
all	1	1	1	1	1
skill	2	2	2	2	2
remove	1	1	1	1	1

O(4)

val=10
val=20

40
30
~~40~~
~~30~~
~~20~~

~~30~~
~~40~~
~~20~~
~~30~~

~~10~~	20	30	40
add 1	1	1	1
	2	2	2
2	2		2
	2		
	2		