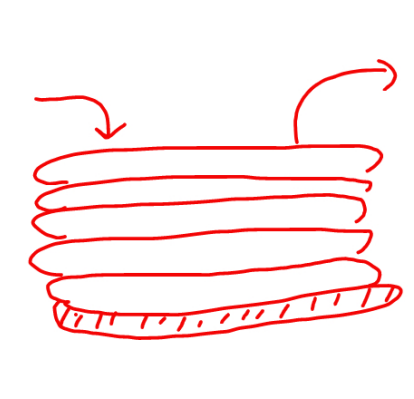
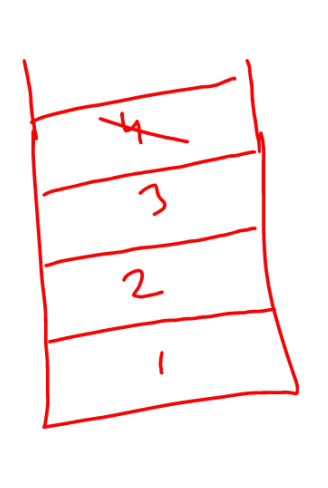
STACKS

(LIFO => Last In First Out)

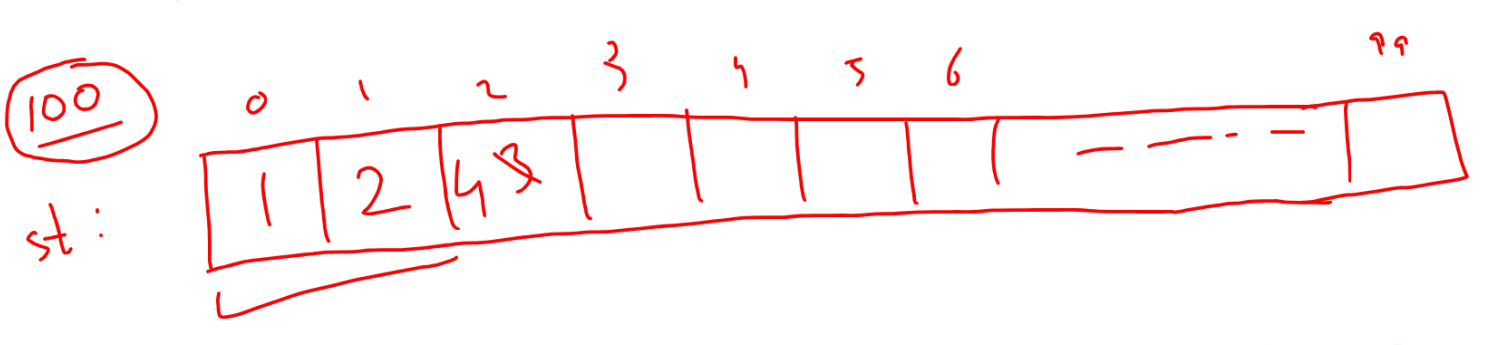




Push(x)
Pop()
top() → 3

st.top()

size = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 2
Push(1)
Push(2)
Push(3)
Pop()
Push(4)



arr[size]

size

head
head head head
st:
3 2 1

(head = head→next)

Push(1)
Push(2)
Push(3)
← Pop()

{ [}] => Unbalanced

[({ })] [ ] balanced

{ [ ] } balanced → True

LIFO

({ [ ] } (

{ { → True

stack

((→ False

"[[((((("

(( )) True

((( )) false.

op: x2
cl: x2

op: x3
cl: x2

Time Complexity: O(len)
Aux. space: O(len)

) (
op: 0
cl: 1
=> False

op < cl → False

[{] } 1/0

1 & (1 1 0)

(1 & (0 & (1 1 1)))

(1 & (0 1 1))

10
11

6
3
0

I/P: [6 8 0 1 3]
O/P: [8 -1 1 3 -1]

i : 0 → n-1
j : i+1 → n-1
O(n²)

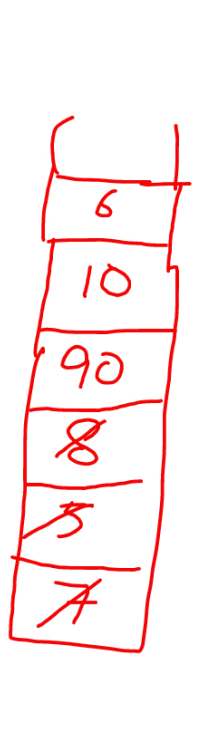
i = ~~0~~ ~~1~~ 2
[j = ~~1~~ ~~2~~ 3]

I/P [6 5 7]
O/P [7 7 -1]

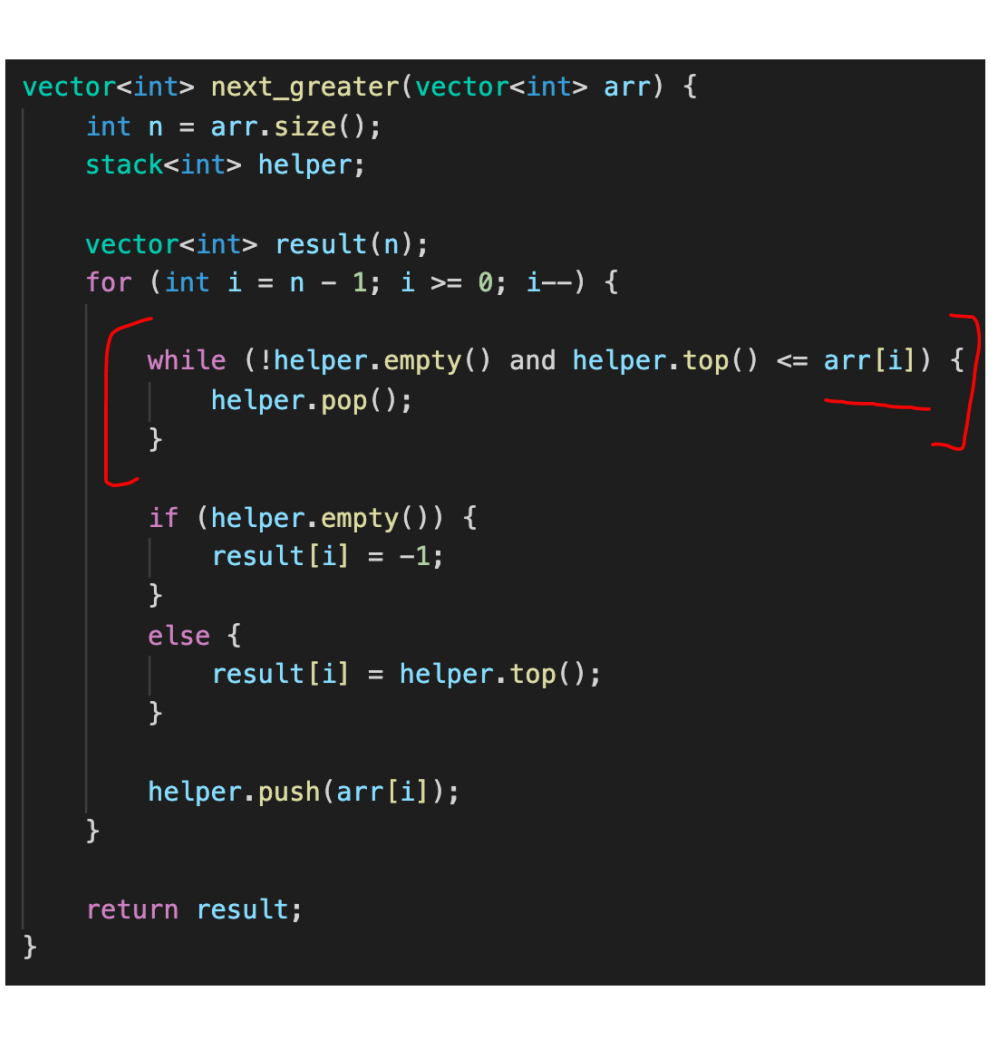
[~~7~~ ~~5~~ 7]

I/P [6 10 90 8 5 7]
O/P = [10 90 -1 -1 7 -1]

O(n)



[TC: O(n)
Aux. space: O(n)]



```
vector<int> next_greater(vector<int> arr) {
    int n = arr.size();
    stack<int> helper;

    vector<int> result(n);
    for (int i = n - 1; i >= 0; i--) {
        while (!helper.empty() and helper.top() <= arr[i]) {
            helper.pop();
        }
        if (helper.empty()) {
            result[i] = -1;
        }
        else {
            result[i] = helper.top();
        }

        helper.push(arr[i]);
    }

    return result;
}
```

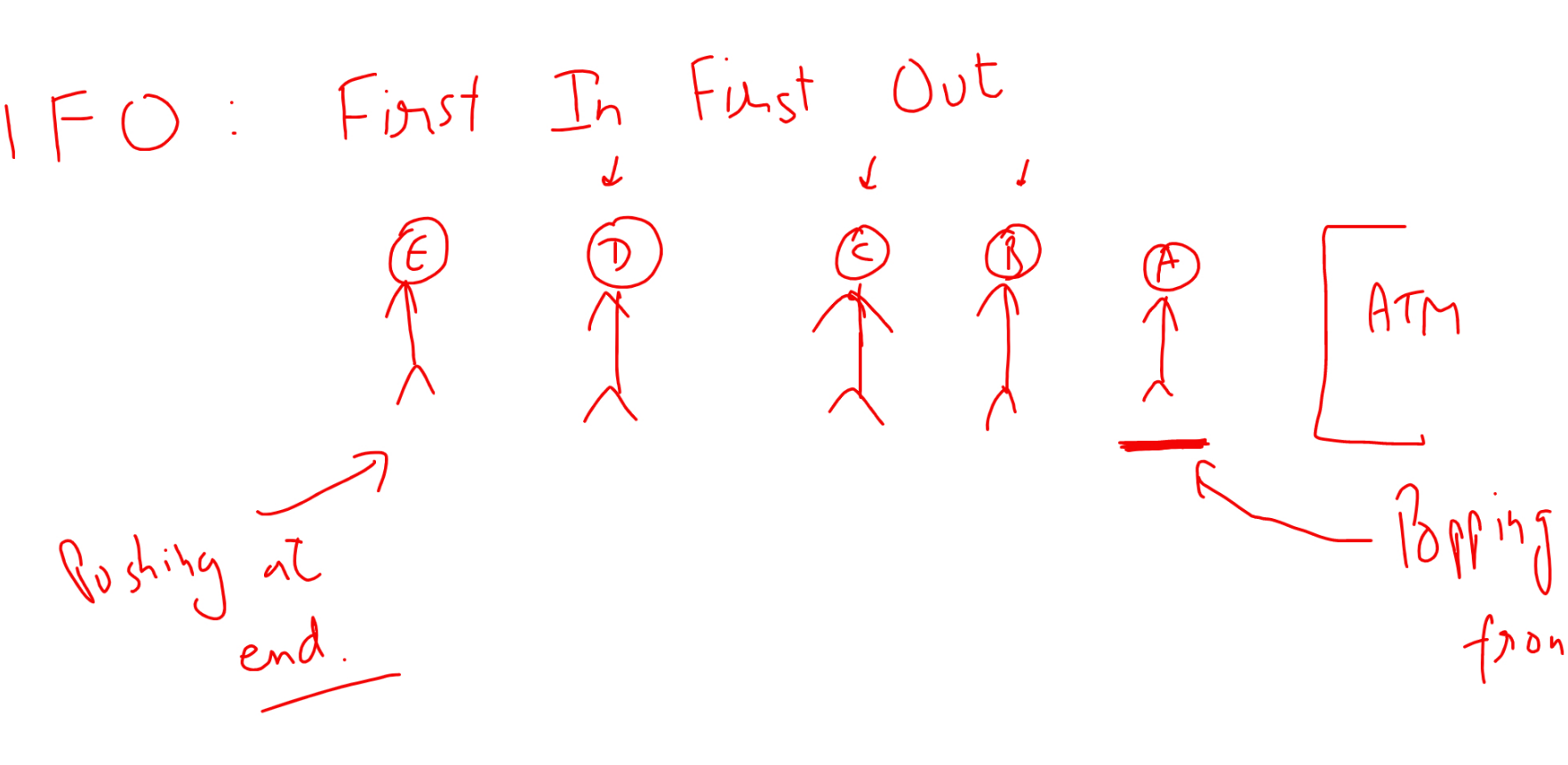
7
[6 10 90 8 5 7]
[10 90 -1 -1 7 -1] ✓

Push(1) Push(2)
[1 | 2]

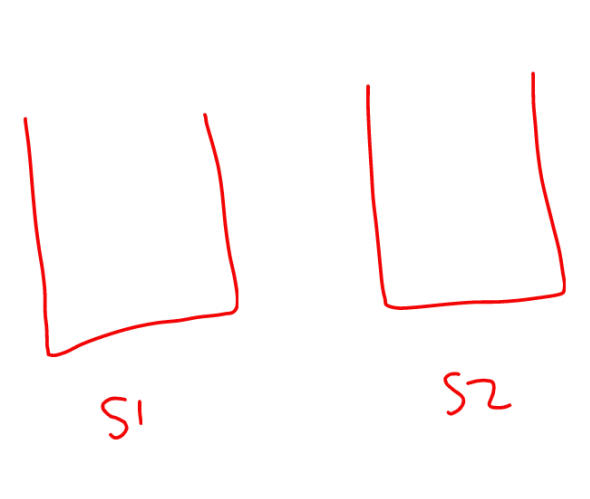
O(1)

QUEUE

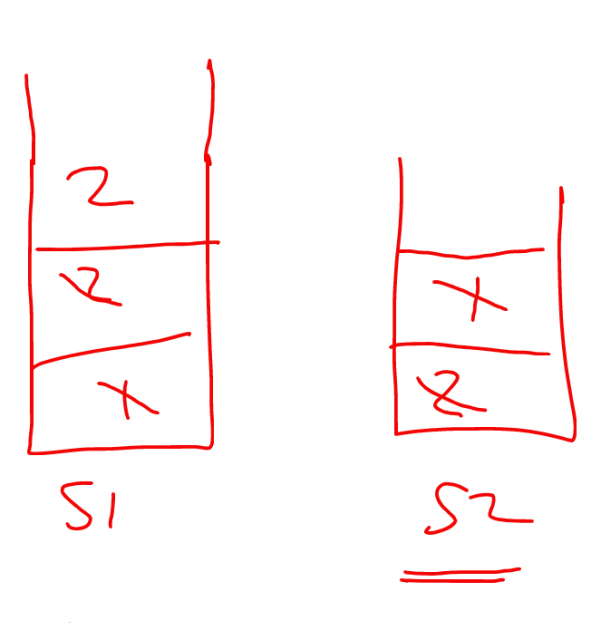
FIFO: First In First Out



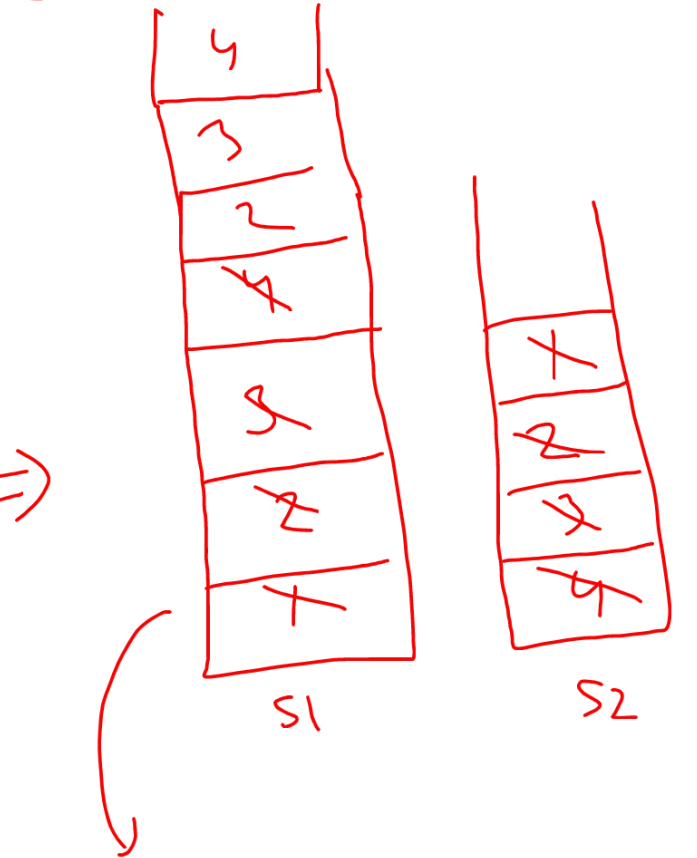
=> Queue_push()
int Queue_pop()



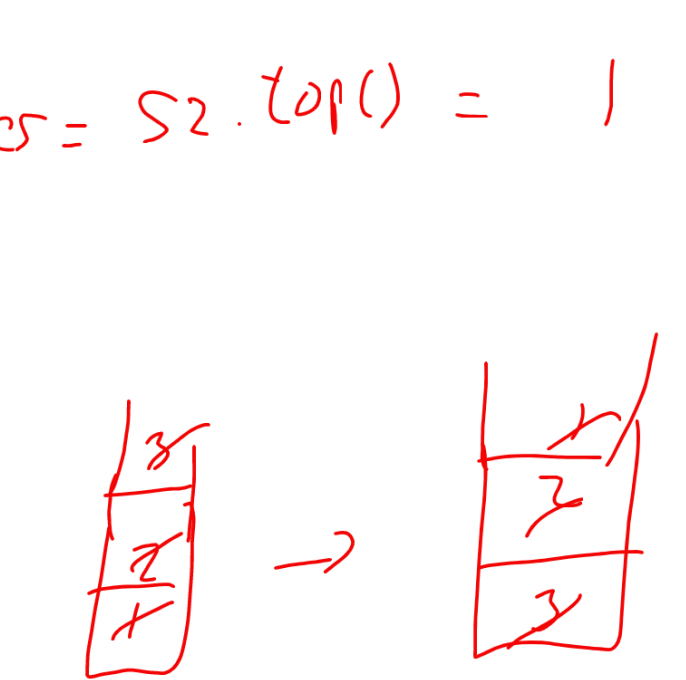
Queue_push(1)
Queue_push(2)
=> Queue_pop() → 1
Queue_push(3)
Queue_pop() → 2

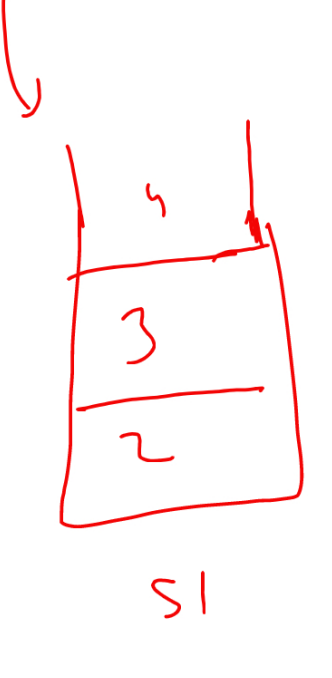


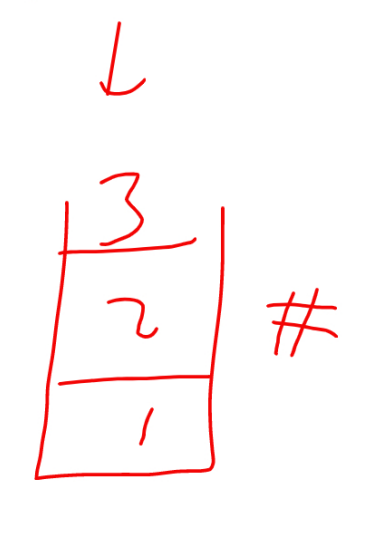
O(1)
O(n)



res = S2.top() = 1







[HW: Implement stack using two queues] #

[HW: Count the sum of set bits in all nos. from 1 to N]

1
10
11
100
101
110
111
1000
1001
1010

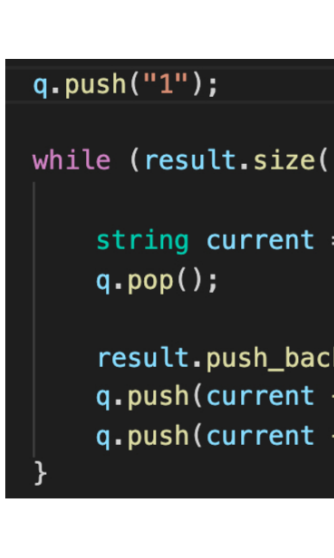
n=10

FIFO

Front → ~~1~~ ~~10~~, ~~11~~, ~~100~~, ~~101~~, ~~110~~, ~~111~~, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111

10

result: 1, 10, 11, 100, 101, 110, 111,



```
q.push("1");

while (result.size() != n) {

    string current = q.front();
    q.pop();

    result.push_back(current);
    q.push(current + "0");
    q.push(current + "1");
}
```

n = 5

queue: front ~~1~~, ~~10~~, ~~11~~, ~~100~~, ~~101~~,
110, 111, 1000, 1001,
1010, 1011

current = ~~1~~
~~10~~
~~11~~
~~100~~
101

result: [1, 10, 11, 100, 101] ✓

[10, 11]

[1 2 2]

vector<vector<int>>

2	2	1	
0	0	0	[ ]
0	0	1	[1]
0	1	0	[2]
0	1	1	[1, 2]
1	0	0	[2]
1	0	1	[1, 2]
1	1	1	[1, 2, 2]

[1, 2, 1] → [1, 1, 2] ✓

1	2	1	
0	0	0	[ ]
0	0	1	[1]
0	1	0	[2]
0	1	1	[1, 2]
1	0	0	[1]
1	0	1	[1, 1]
1	1	0	[2, 1]
1	1	1	[1, 2, 1]

res = [
0 [ ]
1 [ ]
[ ]]

[sort(res.begin(), res.end())]

[___] [___]