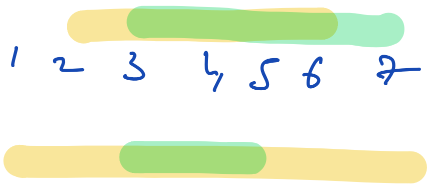
# Q Merge intervals.

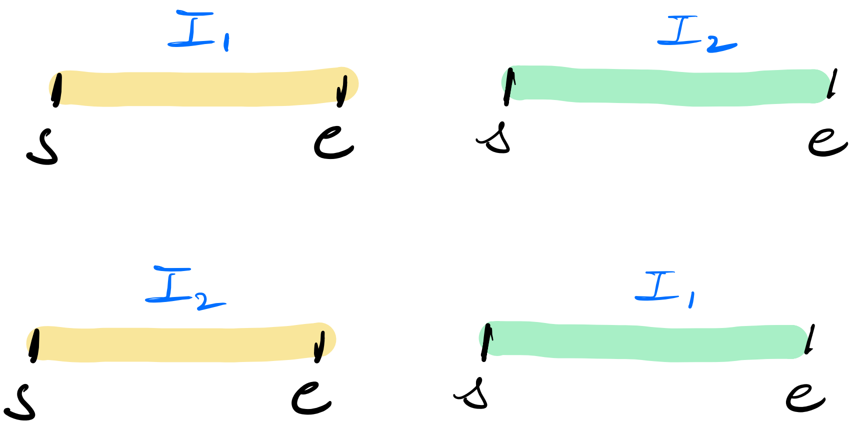
Interval is [a, b]

overlap ⇒ If intersecting at 1 or more elem then they overlap.

Ex

|  | 2 3 4 5 6 | 3 4 5 6 7 |  |
|---|---|---|---|
|  | 2,6 | 3,7 | 2,7 |
|  | 2,8 | 4,6 | 2,8 |
|  | 3,7 | 4,10 | 3,10 |
|  | 3,6 | 6,11 | 3,11 |
|  | 2,5 | 8,10 | NO OVERLAP |

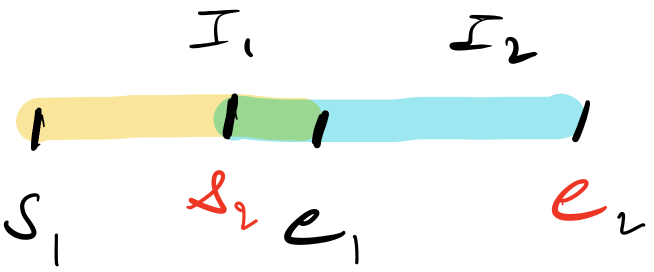




NO OVERLP

if ( $s_2 > e_1$ )

NO overlap



merged interval

s → $\min(s_1, s_2)$

e → $\max(e_1, e_2)$

Cisco, MS, LinkedIN

Given N non overlap intervals, sorted based on start

new interval I comes, merge all

[1,3] I= 12,22 1,3

[4,7] 4,7

[10,14] 10 24 10,24

[16,19] 27,30

[21,24] 32,35

[27,30]

[32,35]

N=5 [1,5] I= [12,22] 1,5

[8,10] 8,10

[11,14] 11 24 11,24

[15,20]

[20,24]

Code

Say ar[N] Intervals , Interval I Comes

```
Intervals [ ] merge (Intervals ar[ ], Interval I){
    for(i=0;i<n ;i++){
        l= ar(i) . start
        r= ar(i) . end
        if( I. start > r) {
            print (ar(i))
        }
        else if ( l > I. end) {
            // all done
            print (I. start, I. end)
            for(j=i ;j<n ;j++ ){
                print (ar(j))
            }
            return
        }
    else {
        I.start = min ( l , I.start)
        I.end   = max ( r, I.end )
    }
}
print ( I )
```

TC: O(N)

[1,3]
[4,7]
[10,14]
[16,19]
[21,24]
[27,30]
[32,35]

1, 3
~~4, 7~~
10, 24
27, 30
32, 35

[1,5]
[8,10]
[11,14]
[15,20]
[20,24]

11, 24
I = [~~12~~, ~~22~~]

1, 5
8, 10
11, 24

```
class Interval {
    int start
    int end
}
```

Q Find first missing natural no.

Unsorted array 1 2 3 4 5 6 - - - -

Eg 1 3 -2 1 2 7 ans = 4

Eg2 -9 2 6 4 -8 1 3

ans = 5

Brute: 1) Sort & check

2) Check for each number starting from 1

key obs : min possible ans = 1

max possible ans = n+1 1 2 3 4 - - - n

Idea : answer can only be from 1 to n+1

we want to solve in SC O(1)

so we want to mark the presence of 1, 2, 3, 4 , - - - -

1 → idx 0 2 → idx 1

How? ⇒ set elem at num-1 idx as -ve

How to handle -ve ?

-ve is useless → yes

replace -ve with useless +ve num

like n+2

-3 1 4 2 6 3

⇒ ?

Now start marking

8 1 4 2 6 3

## Code

```
for( i=0 ;i<N ;i++) {
    if (ar[i] ≤ 0)
        ar[i] = n+2
}

for (i=0 ; i<n ; i++) {
    ele = abs(ar[i])
    if ( ele ≥ 1  && ele ≤ N) {
        idx =  ele -1
        ar[idx] = -1 x ele
    }
}

for (i=0 ;i<N ; i++){
    if (A[i] > 0)
        return i+1
}

return    N+1
```

TC : O(N)

SC : O(1)

Dry run

$n = 8$ $n+2=10$

4 0 1 -5 -10 8 2 6

⇒ num ≤ 0 convert to 10

⇒

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| -4 | -10 | 1 | -10 | 10 | -8 | 2 | -6 |

ans = 3

idx

4 → 3

1 → 0
2 → 1
3 → 2
4 → 3
⋮
x → x-1

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| -5 | -1 | -4 | -2 | -6 | -3 |

-5 -4 -3 -2
6 6 6 6

$n=4$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| -1 | -2 | -3 | -2 | 4 | 2 | 4 |

Q Merge intervals. Sorted by start time

0,2    1,4    5,6    6,8    7,10    8,9    12,14

0,4

5,10

12,14