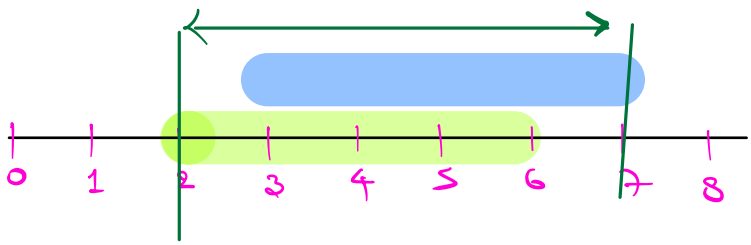
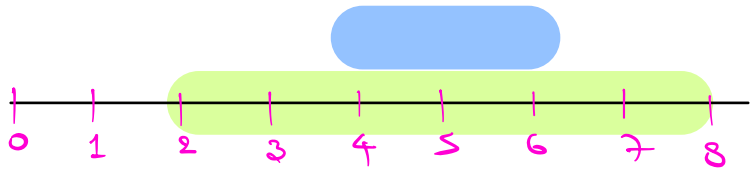
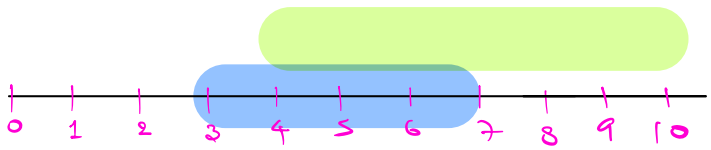
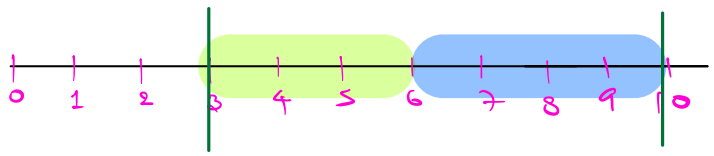
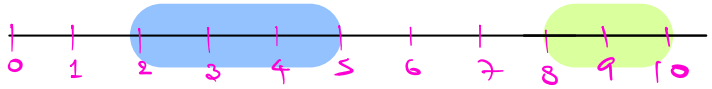
13th & 15th ⇒ No Class.

Merge Intervals

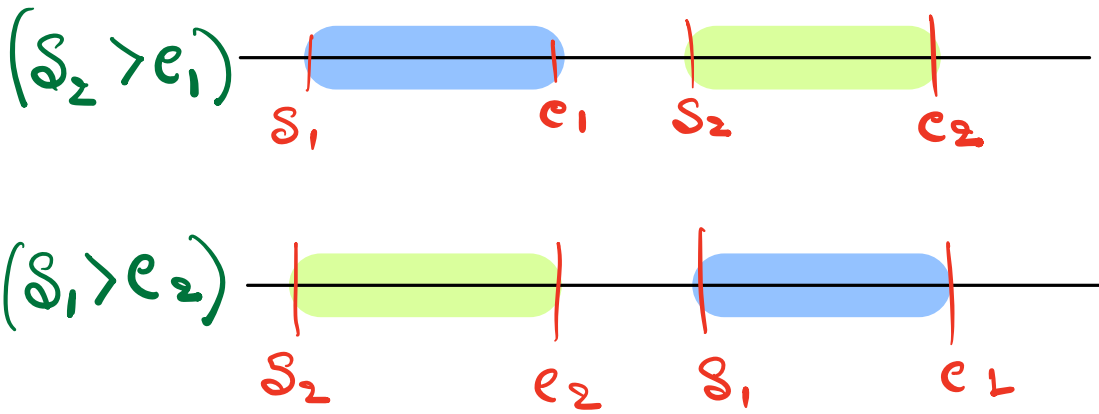
Interval ⇒ start_time & end_time
(s, e)

Intervals

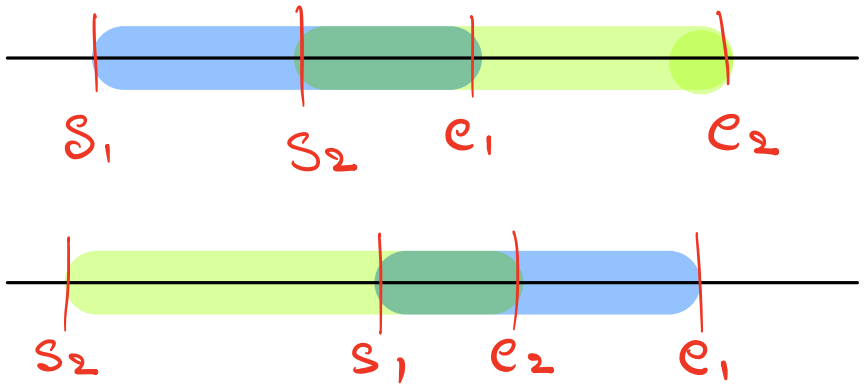
I_1	I_2		Merged Time
(2, 6)	(3, 7)		{2, 7}
(2, 8)	(4, 6)		{2, 8}
(3, 7)	(4, 10)		{3, 10}
(3, 6)	(6, 10)		{3, 10}
(2, 5)	(8, 10)		No Overlap.

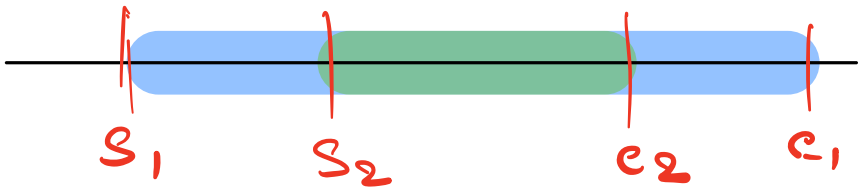
NOTE: To merge 2 intervals they must overlap.

I_1 (s_1, e_1) I_2 (s_2, e_2)




```
if ( s2 > e1  || s1 > e2) {
        Do not overlap
}
else {
        Overlap
}
```





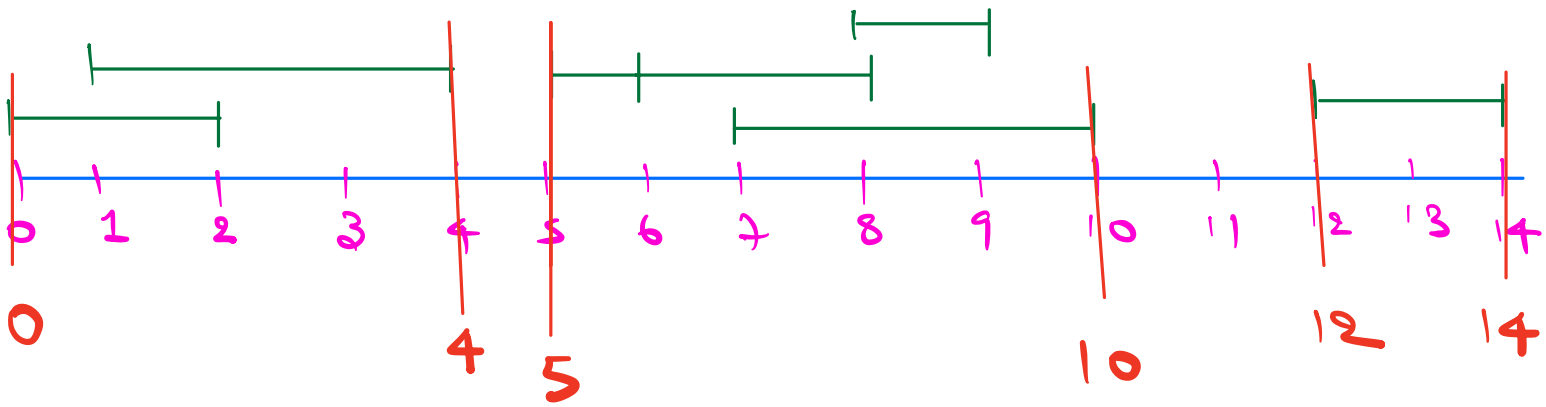
merged start $= \min(s_1, s_2)$

merged end $= \max(e_1, e_2)$

Given a sorted list of overlapping intervals sorted on the basis of start time.

Merge all overlapping intervals & return a final sorted list.

$$A = \begin{array}{c} \\ s \\ e \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 1 & 5 & 6 & 7 & 8 & 12 \\ 2 & 4 & 6 & 8 & 10 & 9 & 14 \end{bmatrix} \end{array}$$



O/p : $\begin{bmatrix} 0, & 5, & 12 \\ 4, & 10, & 14 \end{bmatrix}$

Soln

	0	1	2	3	4	5	6
s	0	1	5	6	7	8	12
e	2	4	6	8	10	9	14

Current	i		Answer List
(0, 2)	(1, 4)	Overlap	
(0, 4)	(5, 6)	No Overlap	(0-4),
(5-6)	(6, 8)	Overlap	(0-4)
(5-8)	(7-10)	Overlap	(0-4)
(5-10)	(8-9)	Overlap	(0-4) (5-10)
(5-10)	(12-14)	No Overlap	(0-4) (5-10)

$e_1 \geq s_2$

Code

class Interval {
int start
int end
}

```
}

List < Interval >.

List < Interval >  ans;

int  currS  =  A[0].start;
int  currE  =  A[0].end;

for (i = 1;  i < A.size();  i++) {
        // Overlap.
        if ( A[i].start <= currE) {
                // Merge
                currS = min (currS, A[i].start);
                currE = max (currE, A[i].end);
        }
        else {
            Interval  temp = new Interval (currS,
                                currE);
            ans.add (temp);
            currS  =  A[i].start;
            currE  =  A[i].end;
        }
}
Interval temp = new Interval (currS, currE);
ans.add (temp);
return ans;
```

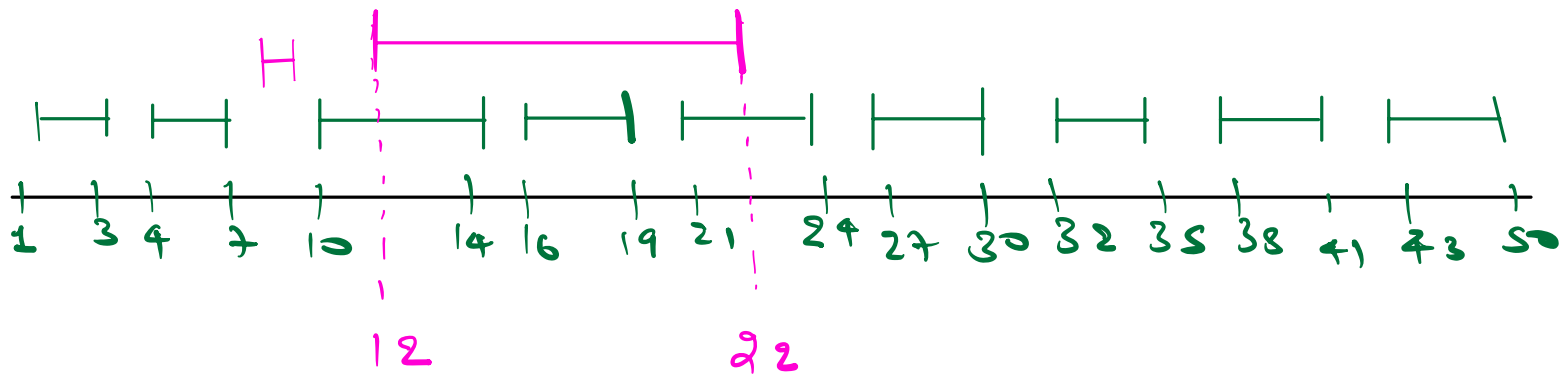
T.C. = O(N)
S.C. = O(1)

Given a sorted list of non-overlapping intervals sorted on the start time

Insert a new given interval such that the final list of intervals is also sorted & non-overlapping.
Print the intervals.

	0	1	2	3	4	5	6	7	8
A =	1	4	10	16	21	27	32	38	43
	3	7	14	19	24	30	35	41	50

New Interval (12, 22)



O/P: [(1, 3), (4, 7), (10, 24), (27, 30), (32, 35), (38, 41), (43, 50)]

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1, & 4, & 10, & 16, & 21, & 27, & 32, & 38, & 43 \\ 3, & 7, & 14, & 19, & 24, & 30, & 35, & 41, & 50 \end{bmatrix}$$

Int (12, 22)

i	New Interval	Overlap	Merge	Print
(1,3)	(12,22)	No.	×	(1,3)
(4,7)	(12,22)	No	×	(4,7)
(10,14)	(12,22)	Yes	(10,22)	×
(16,19)	(10,22)	Yes	(10,22)	×
(21,24)	(10,22)	Yes	(10,24)	×
(27,30)	(10,24)	No		(10-24)
(27,30)				(27-30)
(32,35)				(32,35)
(38,41)				(38,41)
(43,50)				(43,50)

Code

```
void merge ( int Interval[], ns, ne) {

for (i=0; i < N; i++) {
    int L = Interval[i].start;
    int R = Interval[i].end;

if ( ns > R) {
        print (L, R);
```

```
}
else if ( L > ne) {

    print ( ns, ne);
    for ( j = i; j < N; j++) {

        print ( Intervals[j].start,
                Intervals[j].end);

    }
    return;

}
else {
    ns = min ( L, ns);
    ne = max ( R, ne);

}

print (ns, ne);

}
```

T.C. = O(N)
S.C. = O(1)

Given an unsorted array of integers find the first missing Natural no.

A = { 3, -2, 1, 2, 7 } ⇒ 4

$A = \{-9, 2, 6, 4, -8, 1, 3\} \Rightarrow 5$

$A = \{1, 2, 3, 4, 5\} \Rightarrow 6$

$A = \{3, 5, 4, 9, 6\} \Rightarrow 1$

Sol^n) Brute force

Check if all natural no's starting from 1 are present or not.

Code

```
for (i=1; i<=(N+1); i++){
    bool isfound = false;
    for (j=0; j<N; j++){
        if (A[j] == i){
            isfound = true;
            break;
        }
    }
    if (isfound == false)
        print(i);
```

Sieth;

{

}

T.C. = $O(N^2)$
S.C. = $O(1)$

Q If I/P size is N.

What can be the max value of ans.

N ⇒ {1, 2, 3, 4, 5. ... N}

Ans = (N+1)

2) Use Hashset

T.C. = $O(N)$
S.C. = $O(N)$

↓ X

S.C. = $O(1)$

3) Optimised Soln

$$A = \underset{0}{[-1}, \underset{1}{2}, \underset{2}{3}, \underset{3}{4}, \underset{4}{1}, \underset{5}{5}, \underset{6}{7}]$$

$$A = [1, 2, 3, 4, 5, -1, 7]$$

H.W. How to do it

[1, 6]

I/P : [-5, -3, 1, 2, 8, 9]

$1 \rightarrow 0$

$2 \rightarrow 1$

$3 \rightarrow 2$

$\vdots$

$N \rightarrow N-1$