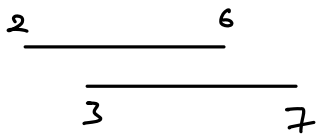
Welcome ☺

Agenda : Interview Problems

Intervals → Merge intervals

overlapping intervals

$[2,6)$ $[3,7)$ $\Rightarrow$ $[2,7)$



$(2,8)$ $[4,6)$ $\Rightarrow$ $[2,8)$

$(3,8)$ $[5,12)$ $\Rightarrow$ $[3,12)$

Non-overlapping intervals

$(5,8)$ $[1,3)$ $\Rightarrow$ no overlap

$\Rightarrow$ Overlapping Interval → (start, end)

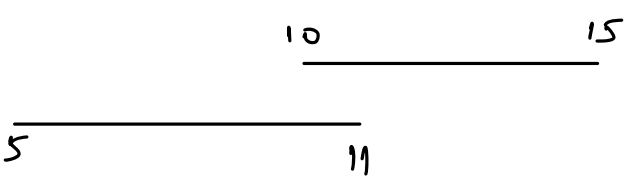
(s_1, e_1) $[s_2, e_2)$

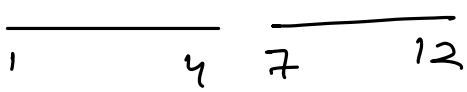
Merged interval → $\min(s_1, s_2)$, $\max(e_1, e_2)$

How to check if intervals are overlapping

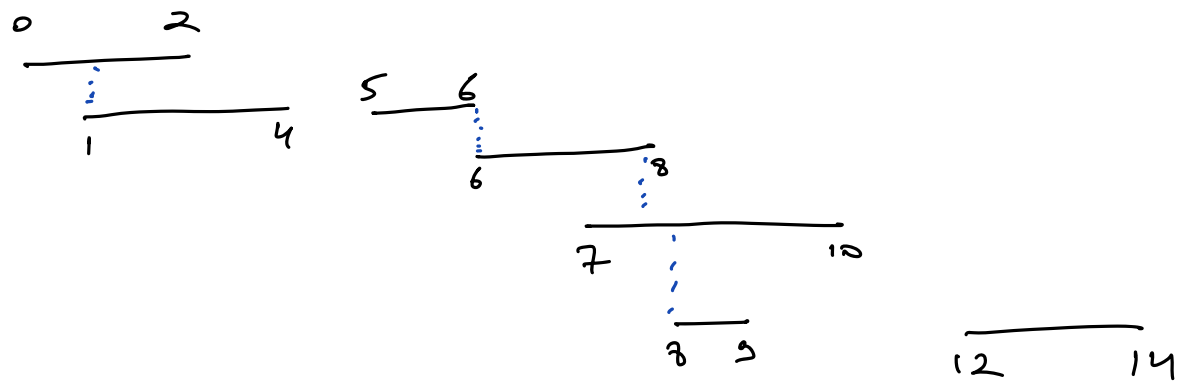
$$E_1 \geq S_2 \text{ AND } E_2 \geq S_1$$ ✓

eg: $(10,15)$ $[5,11)$ $(1,4)$ $[7,12)$





Q Given a sorted list of overlapping intervals, sorted based on start time. Merge all overlapping intervals and return sorted list.



⇒ (0,4) (5,10) (12,14)

Overlapping condⁿ ⇒ $E_1 \geq S_2$

Merged intervals ⇒ S → S1

E ⇒ max(E1, E2)

Code

```
L = S[0]      R = E[0]

for( i = 1 ; i < N ; i++)
{
    if ( R >= S[i] ) // overlapping
    {
        R = max( R, E[i] )
    }
    else
    {
        print ( L, R)
        L = S[i]
        R = E[i]
    }
}
print ( L, R)
```

T.C ⇒ O(N)

S.C ⇒ O(1)

Q Given an unsorted array of integers. Find the first missing natural number.

eg: 3 -2 1 2 7 ⇒ 4

1 0 -5 -6 4 2 ⇒ 3

Brutforce ∀ natural no., check if they are present in array

T.C ⇒ O(ans) * O(N)

O(N * N) ≃ O(N^2)

App2 Sort

3 -2 1 2 7

!!

-2 1 2 3 7

T.C ⇒ O(N log N)

App.3

0 1 2 3 4 5 6 7 ↓

eg: [-5, 3, 10, 8, 1, 2, 4, -3]

vst [F F F F F F F F]

T T T T T

missing = 5

T.C ⇒ O(N+N) ≃ O(N)

S.C ⇒ O(N)

Q Next Permutation

Rearrange numbers into next numerically greater permutation.
If not possible, then return lowest possible no.

eg: [1 2 3]
⇒ [1 3 2]

eg: [3 2 1]
⇒ [1 2 3]

1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1 ↓

eg: [3 1 2]
⇒ [3 2 1]

eg [1 2 3 6 5 4]
⇒ [1 2 4 6 5 3]

S1) $s[i] < s[i+1]$ right to left. →

S2) Find smallest among the lot which is JUST greater than the element which will be swapped.

S3 Rearrange the remaining array in sorted order s.t we JUST the larger number

eg: 3 1 6 4 2 0

o/p

3 0 6 4 2 1

3 0 1 2 4 6 ✗

3 2 6 4 1 0

⇒ 3 2 0 1 4 6 ✓

H.W

T.C ⇒ O(N)

3 2 1

⇒ 1 2 3 ✓ Reverse the array.

⇒

Reattempt 1 → 18th → 25th Oct

" 2 → 25th → 1st Nov

" 3 → 1st Nov → 16th Nov.