Arrays - 1

Q⇒ Given an integer array
Find the max value of

$f(i,j) = |A[i]-A[j]| + |i-j| \quad \forall (i,j)$ $|x| = abs(x)$

Eg ⇒ A = [1, 3, -1] $j-i$

i	j	f(i,j)
0	0	|1-1| + |0-0| = 0
0	1	|1-3| + |0-1| = 3
0	2	|1-(-1)| + |0-2| = 4
1	0	|3-1| + |1-0| = 3
1	1	|3-3| + |1-1| = 0
1	2	|3-(-1)| + |1-2| = ⑤
2	0	|-1-1| + |2-0| = 4
2	1	|-1-3| + |2-1| = ⑤
2	2	|-1-(-1)| + |2-2| = 0

T.C. = $O(n^2)$ max = -∞

```
for (i=0 to N)
    for (j=0 to N)
        f(i,j)
        if (f(i,j) > max)
            update max;
```

S.C. = O(1)

Brute force ⇒ Calculating all possibilities to find the answer

Observations

1) f(i, i) = 0   i = j

2) f(i,j) = f(j,i) ⇒ Computing absolute values

(1, 2) (i<j) (2, 1) (i>j)

⇒ Obs. 2 concludes that we can calculate for only (i<j)

$|(i-j)| = -ve$

$f(i,j) = |A[i]-A[j]| + (j-i)$

$|x| \to x>0 = x$, $x<0 = -x$

Case I if (A[i] ≤ A[j])

$f(i,j) = A[j]-A[i] + (j-i)$

$= \underbrace{(A[j]+j)}_{x_j \text{ max}} - \underbrace{(A[i]+i)}_{x_i \text{ min}}$ $\forall i<j$

```
for (i=0 to N-1)
    for (j=i+1 to N-1)
```

Case II if (A[i] > A[j])

$f(i,j) = A[i]-A[j] + (j-i)$

$(A[i]-i) - (A[j]-j)$

(i, j)

A = [0, 50, 1, 0] (indices 0 1 2 3)

A = [3, 3, 7, 6, 1, 3] (indices 0 1 2 3 4)

|7-3| + |2-0| = 6 (2-1)

6-3 + 3 = 6

Ans = Max ((A[j]+j) - (A[i]+i), (A[i]-i) - (A[j]-j)) $\forall i<j$

Case I

$X_i = A[i] + i$

Ans = Max $\forall_{ij}$ $(X_j - X_i)$ (max, min) O(n)

A ⇒ [ ]
X ⇒ [ ] > O(n)

Case II

$Y_i = A[i] - i$

Y = max $(Y_j - Y_i)$ (max, min)

Overall Ans = Max $(X_{max} - X_{min}, Y_{max} - Y_{min})$

Eg ⇒ A = [1, 3, -1]   O(1)

O(N) ⇒ X = [1, ④, ①] $X_{max} = 4$, $X_{min} = 1$ O(N)

O(N) ⇒ Y = [1, 2, -3] $Y_{max} = 2$, $Y_{min} = -3$ O(N)

= Ans = Max(4-1, 2-(-3))

Max (3, 5) = ⑤

T.C. = O(N)

S.C. = O(N) ⇓ O(1)

```
for (i=0 to N)
    X = A[i]+i
    Y = A[i]-i
    if (X > Xmax) {
        Xmax = X; }     > Xmin, Ymax, Ymin
```

Q⇒ Given an integer array find the first +ve integer missing in the array

Eg ⇒ A ⇒ [8, 10, ①, -3, ②, -5] 4

Ans = 3

Soln

1) Sort the array

[-5, -3, ①, 2, 8, 10]

>1 ⇒ 1 is the answer

Ans = 3

T.C. = O(nlogn)
O(nlogn + n) ⇓ sorting

Quick Sort

S.C. = O(1)

2) A = [8, 10, 1, -3, 2, -5] -48 6
  1 2 3 4 5 6

P ⇒ [F | 0, 1, 2, 3, 4 . . . . N+1 7

I Max of the array

II N+1 ⇒ N size of i/p array N

A = [~~8~~, ~~10~~, ~~1~~, ~~-3~~, 2, ~~-5~~, ~~0~~] O(1)

P = [T T T F F F F F]
  0 ① ② ③ 4 5 6 7

O(1)

N+1 (N+1)

min = 1

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

[8 7 0 2 3 6 1 4]

Soln 3 $i \Rightarrow i-1$

A = [~~8~~, ~~10~~, ~~1~~, -3, ~~2~~, -5] N=6
  1 2 8 10
swap(i, A[i]-1)

[1, N] max ans = 7

⇒ [1, 2, 3, 4, 5, 6]
  0 1 2 3 4 5

[1, 2, 8, -3, 10, -5]
  0 1 ② 3 4 5
  i+1 = 3

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

T.C. = O(N)

S.C. = O(1)

A ⇒ [4, 7, 3, 1, 2, 8, 2] ⇒ min +ve ⇒ new +ve value
  0 1 2 3 4 5

A ⇒ [1, 2, 3, ⑦, 8, 6] 4

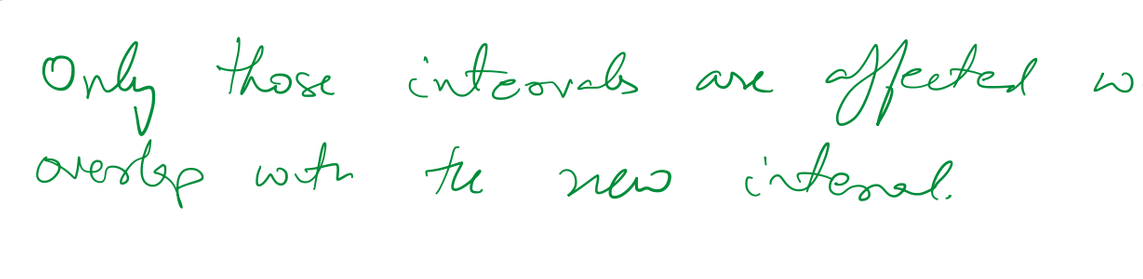
A[i] != (i+1)

Q⇒ Given a sorted set of non overlapping intervals
Insert a new interval s.t. final list is also sorted and non overlapping.

Eg ⇒ S ⇒ [1, 5, 10, 13, 17]
E ⇒ [3, 9, 11, 16, 20]

for any interval i   S[i] ≤ E[i]
Start = S[i]   E[i] < S[i+1]
End = E[i]

Insert l = 8 r = 12

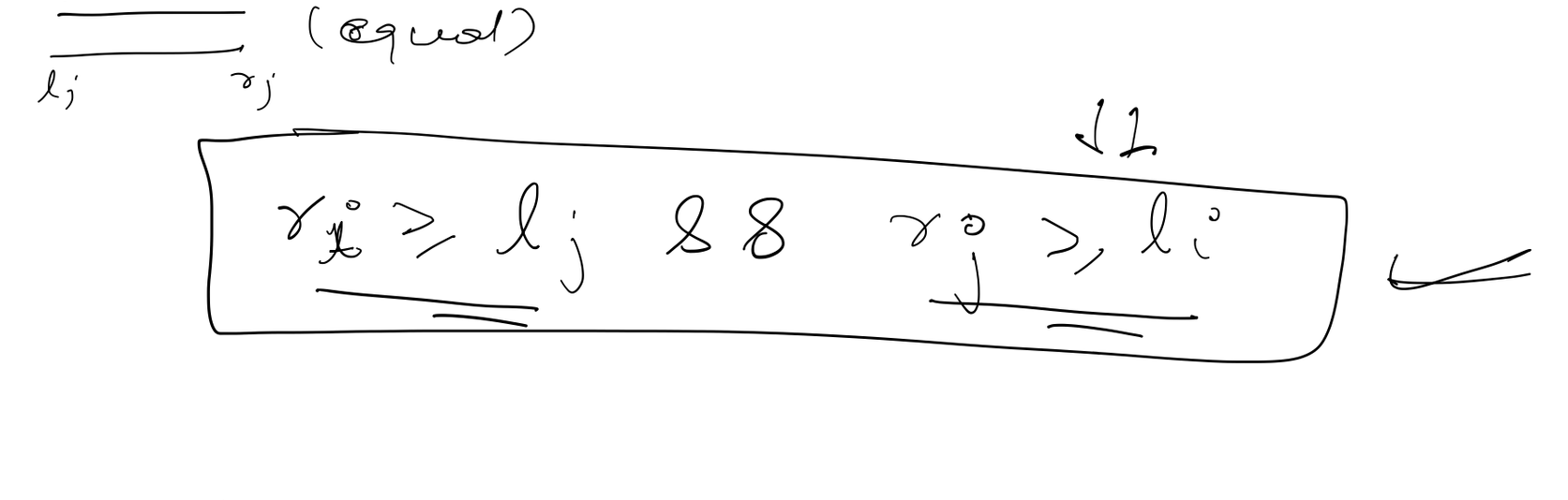


S ⇒ [1, ⑤, 13, 17]
E ⇒ [3, ⑫, 16, 20]

Observations

1) Only those intervals are affected which overlap with the new interval.

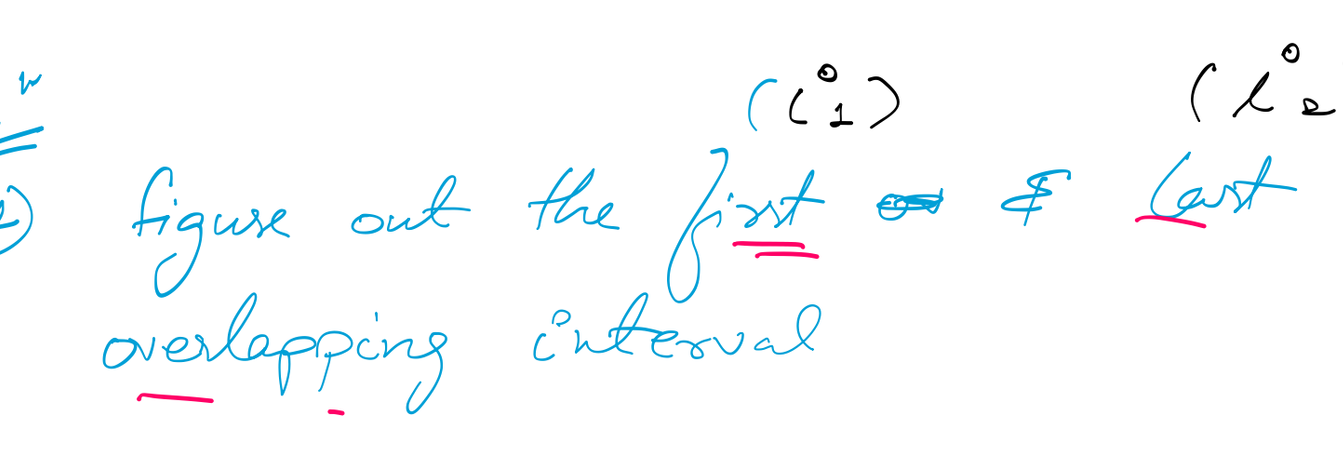
⇒ If interval 1 (l_i, r_i) & interval 2 (l_j, r_j)



$\boxed{r_i \geq l_j \;\&\&\; r_j \geq l_i}$

Q⇒ Start & end of the interval to add.

100 consecutive intervals



Soln
1) figure out the first & last overlapping interval (i_1) (i_2)

Start ⇒ min(S[i₁], start of new interval)

End ⇒ max(E[i₂], end time of new interval)

④

T.C. = O(n)

S.C. = O(1)

new o/p array

S [ ]

S = [1, 5, 10, 14]

e = [3, 9, ⑬, 16]