Today's **Agenda** :-

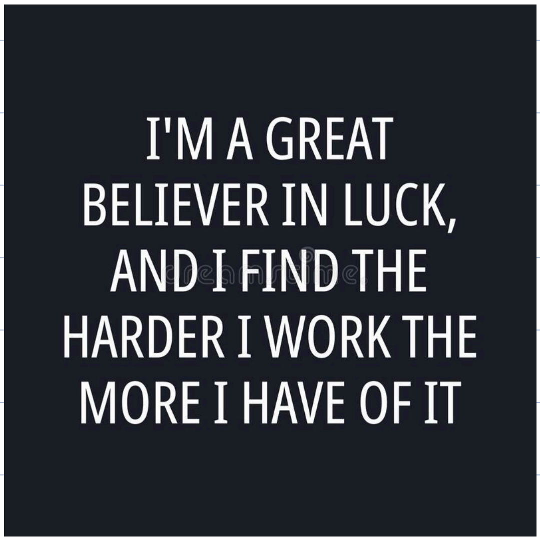
→ Product array puzzle

→ Max Consecutive 1's

→ No. of triplets

Problem Solving Session **tomorrow** : 1-3pm

Today's **Quote** !-



I'M A GREAT BELIEVER IN LUCK, AND I FIND THE HARDER I WORK THE MORE I HAVE OF IT

Ques) Max Consecutive 1's.

Given a $\overset{0 \text{ or } 1}{\text{Binary}}$ array, we can atmost replace a single 0 with 1, find max consecutive 1's we can get.

Ex 1) $\underset{\substack{\downarrow\\4}}{0}\ 1\ 1\ 1\ \underset{\substack{\downarrow\\6}}{0}\ 1\ 1\ \underset{\substack{\downarrow\\5}}{0}\ 1\ 1\ \underset{\substack{\downarrow\\3}}{0}$

Ex 2)

$1\ 1\ 1\ \underset{\substack{\downarrow\\8}}{0}\ 1\ 1\ 1\ 1\ \underset{\substack{\downarrow\\7}}{0}\ 1\ 1\ \underset{\substack{\downarrow\\3}}{0}\ \underset{\substack{\downarrow\\4}}{0}\ 1\ 1\ 1\ \underset{\substack{\downarrow\\6}}{0}\ 1\ 1$

Ex 3) 1 1 1 0 1 1 0 1 1 1 0 1

⇒ 6

Ex 4) 1 1 1 1 1 → 5

: Brute force :-

```
ans = 0
// Iterate & get count of 1's
if (cnt == n) { return n }

for (i=0; i<N; i++) {
    if (arr[i] == 0) {
        l = 0;
        (j = i-1; j>=0; j--) {
            if (arr[j]==1) { l++ }
            else { Break }
        }
        r = 0;
        (j = i+1; j<n; j++) {
            if (arr[j]==1) { r++ }
            else { Break }
        }
        T = l+r+1
        ans = max(ans, T);
    }
}
```

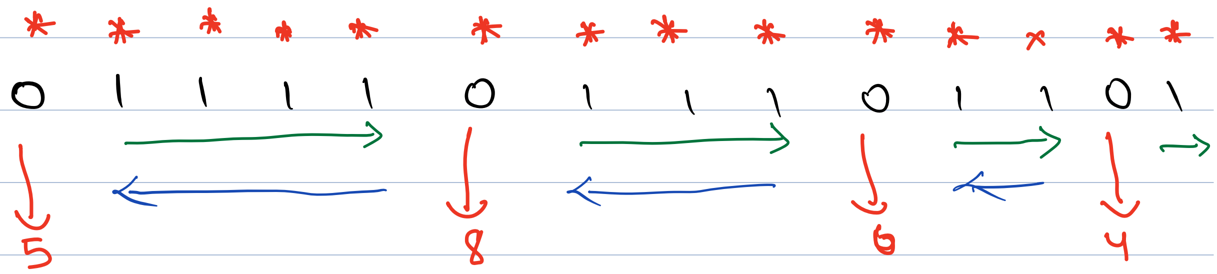
T.C → $O(N^2)$ ✗ → T.C → $O(N)$

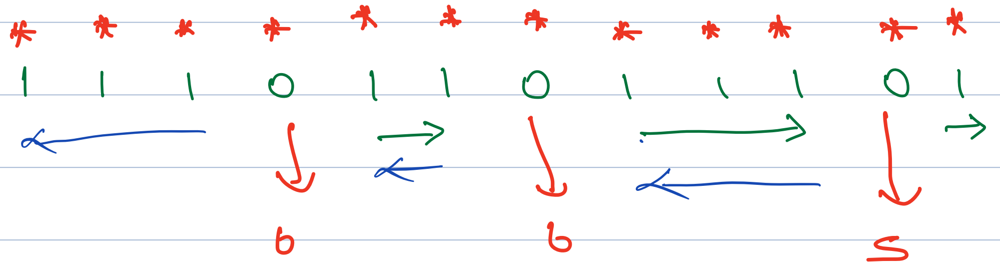
S.C → $O(1)$

```
for (j=1; j<=3; j++){
    for (i=0; i<n; i++){
        print (arr— )
    }
}
```

Iteration → 3n

T.C → O(n).





T.C → O(n).

Ques) Given a Binary array, we can atmost swap a single 0 with 1, find maxm consecutive 1's we can get.

Ex1:-

1 1 0 1 1 0 1

↓ 5 L+R+1

↓ 4 L+R+1

Ex2

1 1 0 1 0 1 0 1 1

↓ 4 L+R+1

↓ 3 ↓ L+R+1

↓ 4 ↓ L+R+1

Ex3)

1 1 1 0 1 1 1

↓ L+R

Ex4)

1 1 0 1 1 1 0 0

↓ 5 L+R

↓ 4 L+R+1

↓ 1 L+R+1

```
ans = 0
// Iterate & get count of 1's
if (cnt == n) { return n }

for (i=0; i<N; i++) {
        if (arr[i] == 0) {
                l = 0;
                (j = i-1; j>=0; j--) {
                    if (arr[j] == 1) { l++ }
                    else { Break }
                }
                r = 0;
                (j = i+1; j<n; j++) {
                    if (arr[j] == 1) { r++ }
                    else { break }
                }

                if (cnt > l+r) {
                    ans = max(ans, l+r+1)
                } else {
                    ans = max(ans, l+r)
                }
        }
}
```

9:58 - 10:03   5 mins break

Ques) find no of triplets i, j, k s.t.
$i < j < k$, && $arr[i] < arr[j] < arr[k]$

Ex 1)

0	1	2	3	4
3	4	6	9	2

Triplets

i	j	k
0	1	2
0	2	3
0	1	3
1	2	3

→ 4

Ex 2)

0	1	2	3	4
2	6	9	4	10

2 , 6, 9, 4, 10

i,	j	k
0	1	2
0	1	4
0	2	4
1	2	4
0	3	4

} → 5

Brute force :-

```
for (i=0; i<n; i++){
    for(J=i+1; J<n; J++){
        for(k=J+1; k<n; k++){

        }
    }
}
```

T.C → $O(n^3)$

S.C → $O(1)$.

$i < j < k$

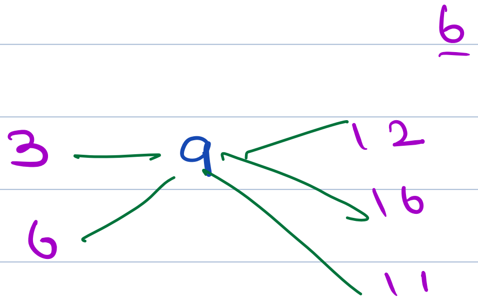
$arr[i] < arr[j] < arr[J]$

Eg:-

J

3 6 9 12 5 16 8 7 11

↑ ↑ ↑ ↑ ↑



```
cnt = 0;
for(j = 1;j < n-1;j++){
    j is middle element,
    left = no. of element less than arr[j]
                                  [0,j-1]

    right = no. of elements greater
                  than arr[j], [j+1, n-1]

    cnt = cnt + (left * right)

}
```

T.C → $O(n^2)$.

S.C → $O(1)$.

Complete Code

↓

T.C → $O(n \log n)$

1) Balanced BST
2) Segment Tree

```
cnt = 0;
for (j = 1 ; j < n-1 ; j++) {
    j is middle element,
        left = 0;
        for (i = j-1 ; i >= 0 ; i--) {
                if (arr[i] < arr[j])
                {
                    left++
                }
        }

        right = 0;
        for (i = j+1 ; i < n ; i++) {
                if (arr[i] > arr[j])
                {
                    right++
                }
        }

        cnt = cnt + (left * right)

}
```

Prefix is not possible.

Break :- 10:42 : 10:50 pm

→ After class

→ weekend

→ Months

Ques

1Q) Given N Array elements replace every element

→ ar[i] with prod of all array elements except itself

→ You cannot / in your code?

Ex1) { 2, 4, 1, 3, 5 }

60 30 | 40 24

↓

120

Ex2) { 1, 6, 2, 3 } → 36

36 6 18 12

Pseudo Code :-

```
for (i=0; i<n; i++) {
    left = Product of all elements
                        till i-1
    right = product of all elements
                  i+1 to n-1

    prod[i] = left * right
}
```

T.C → $O(n^2)$

S.C → $O(1)$

{2, 4, 1, 3, 5}

Prf → 2 8 8 24 120

Sfp → 120 60 15 15 5

```
    prp[0] = arr[0]
else
    prp[i] = prp[i-1] * arr[i]
```

i → prp[i-1] * sfp[i+1]

T.C → (N + N + N) → O(N)

S.C → O(N)

1 1 1 0 1 1 0 1 1 0

ls = ~~0~~ ~~8~~ 2

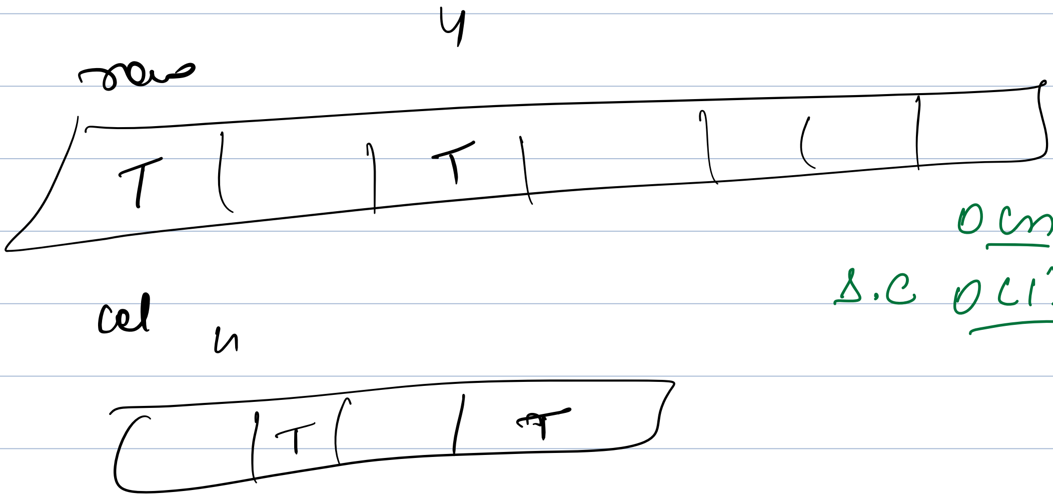
rs = ~~0~~ ~~1~~ ~~0~~ ~~0~~ ~~2~~ 0 2

max = 0

{2, 4, 1, 3, 5}

sfp 120 60 15 15 5

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



O(n)

S.C O(1)