Learnings from previous sessions

⇒ Prefix Sum → Range Queries [eg. sum from index s to e]

⇒ Carry Forward Technique → using previously calculated result

⇒ Subarrays → contiguous part of an array

* Sliding window → answer w.r.t to the given window size

* Contribution by an A[i] to the answer
  → sum of all subarray sums.

⇒ 2D arrays

Que 1. Given an integer matrix [N][N], print all boundaries in clockwise direction

LinkedIn | Amazon | Freecharge | Airtel

	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

Take 4 loop

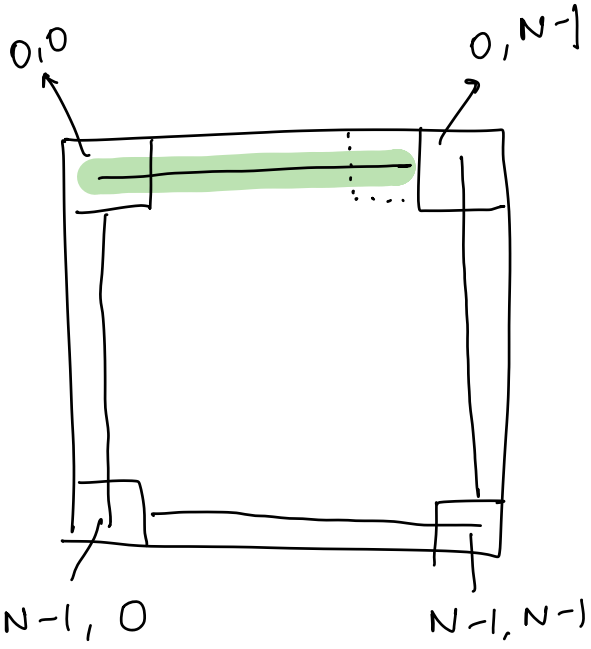
// First row
i → 0
j → [0 N-1]

// last col
i → [1 N-1]
j → N-1

// last row
i → N-1
j → [N-2 0]

// first column
i → [N-2 1]
j → 0

1 6 11 16 21 22 23 24 25 20 15 10 5 4 3 2



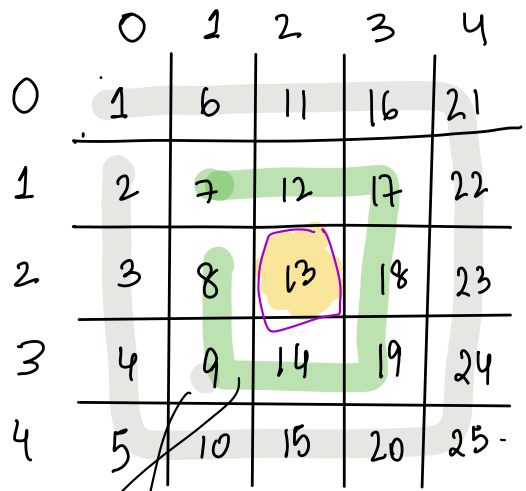
	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

In each loop, we'll print N-1 elements

```
i=0  j=0
while(N>1){
    for(K=1; K<=N-1; K++){
        print(A[i][j])
        j++
    }
    for(K=1; K<=N-1; K++){
        print(A[i][j])
        i++
    }
    for(K=1; K<=N-1; K++){
        print(A[i][j])
        j--
    }
    for(K=1; K<=N-1; K++){
        print(A[i][j])
        i--
    }
    N=N-2
    i++
    j++
}
if(N==1) print(A[i][j])
```

K	i	j
	0	0
1	0	1
2	0	2
3	0	3
4	0	4
5		
	0	4
1	1	4
2	2	4
3	3	4
4	4	4
5		
	4	4
1	4	3
2	4	2
3	4	1
4	4	0
5		
	4	0
1	3	0
2	2	0
3	1	0
4	0	0
5		

0 0

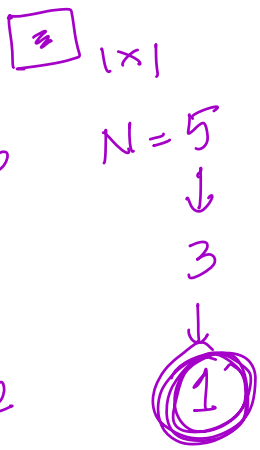


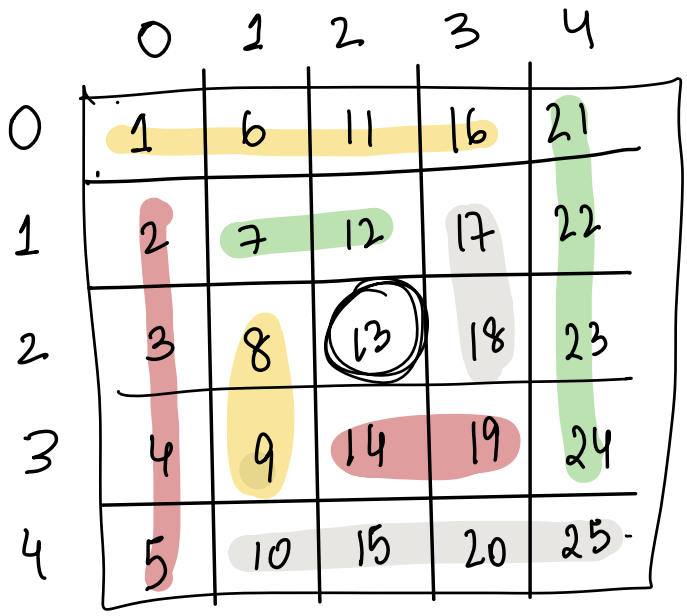
same problem with smaller size

$N = N - 2$

$i++$

$j++$





$TC \rightarrow O(N^2)$

$SC \rightarrow O(1)$

$N = 5$

$i = 0 \quad j = 0$

$i = 0 \quad j = 4$

$i = 4 \quad j = 4$

$i = 4 \quad j = 0$

$i = 0 \quad j = 0$

$N = 5 - 2 = 3$

$i++ \rightarrow 1$

$j++ \rightarrow 1$

$i = 1 \quad j = 3$

$i = 3 \quad j = 3$

$i = 3 \quad j = 1$

$i = 1 \quad j = 1$

$N = N - 2 = 3 - 2 = 1$

$i = 2$

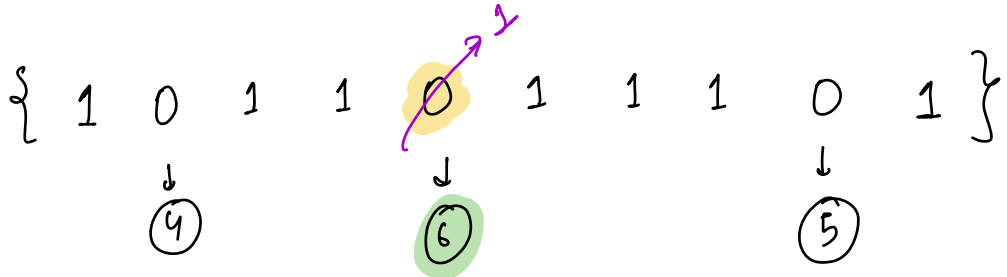
$j = 2$

Q2. Max Consecutive 1s in an array. Amazon

Given a 1D array containing just 1s & 0s.

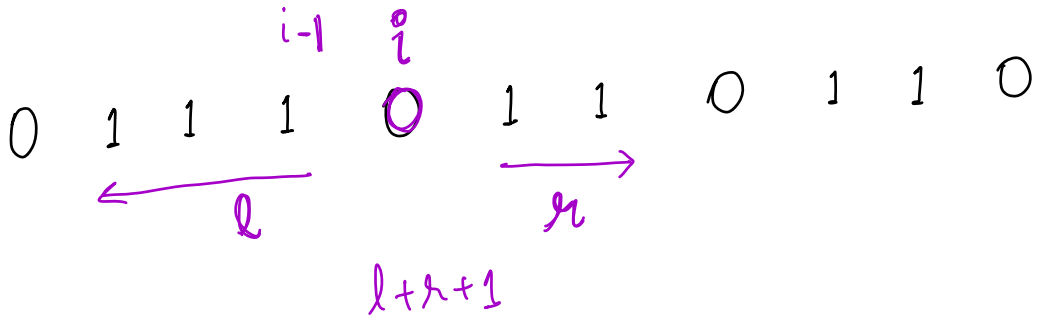
Find the length of max consecutive 1s.

* You can replace atmost one 0 with 1.

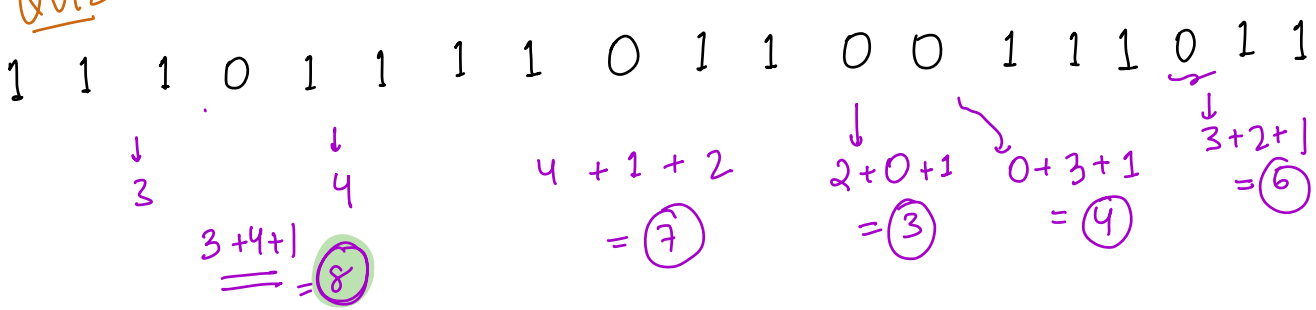


Ans → 6

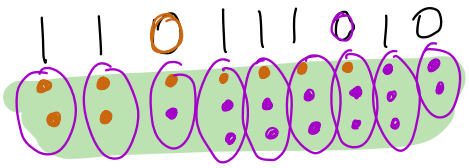
— QUIZ —



QUIZ



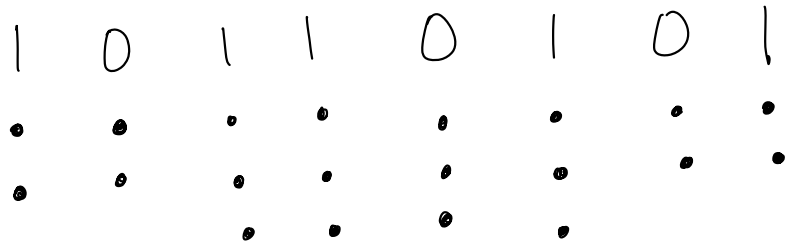
```
bool flag = false
ans = 0
for(i = 0 ; i < N ; i++) {
    if ( A[i] == 0){
        flag = true
        // l = consecutive count of 1s on left
        l = 0
        for( j = i-1 ; j >= 0 ; j--){
            if( A[j] == 1) l++
            else break
        }
        // r = consecutive count of 1s on right
        r = 0.
        for ( j = i+1 ; j < N ; j++){
            if (A[j] == 1) r++
            else break
        }
        ans = max ( ans, l+r+1)
                ↳ if ( l+r+1 > ans){
                      ans = l+r+1
                  }
    }
}
if ( flag == false ) return N
return ans
```



What if all 1s are present ?

1 1 1 1 1 → (N)

DRY RUN



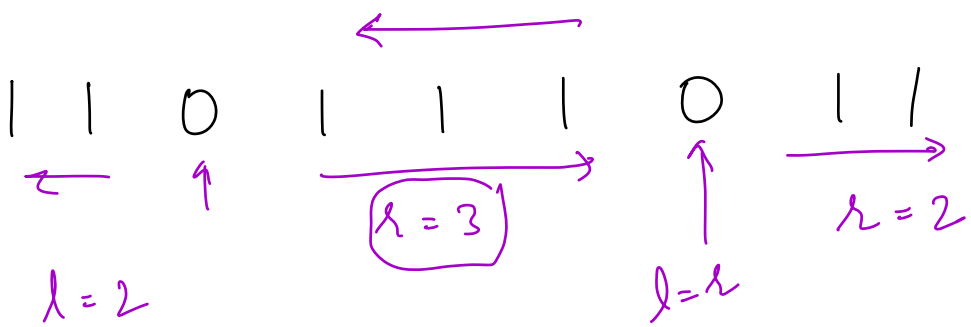
We are visiting each element at max thrice.

Iteration : 3 X N [at max]

TC → O(N)

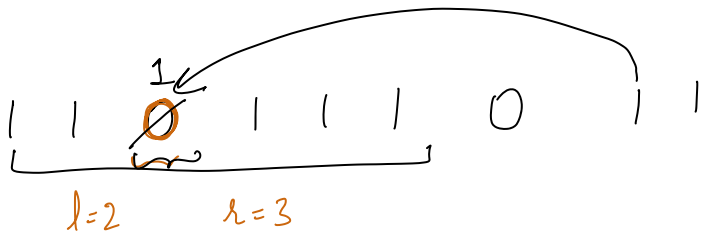
HW.

Approach with less iterations but with same TC.



Modified version of Max Consecutive 1s.

swap atmost one 0 with 1 present in array itself.



2+3+1

(6)

l=1 r=1

Eg 1 1 0 1 0 1 0 1 1

l+r

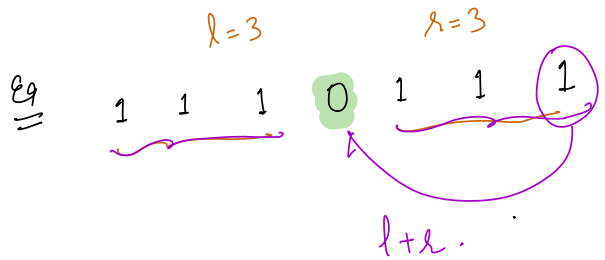
l+r < total → l+r+1

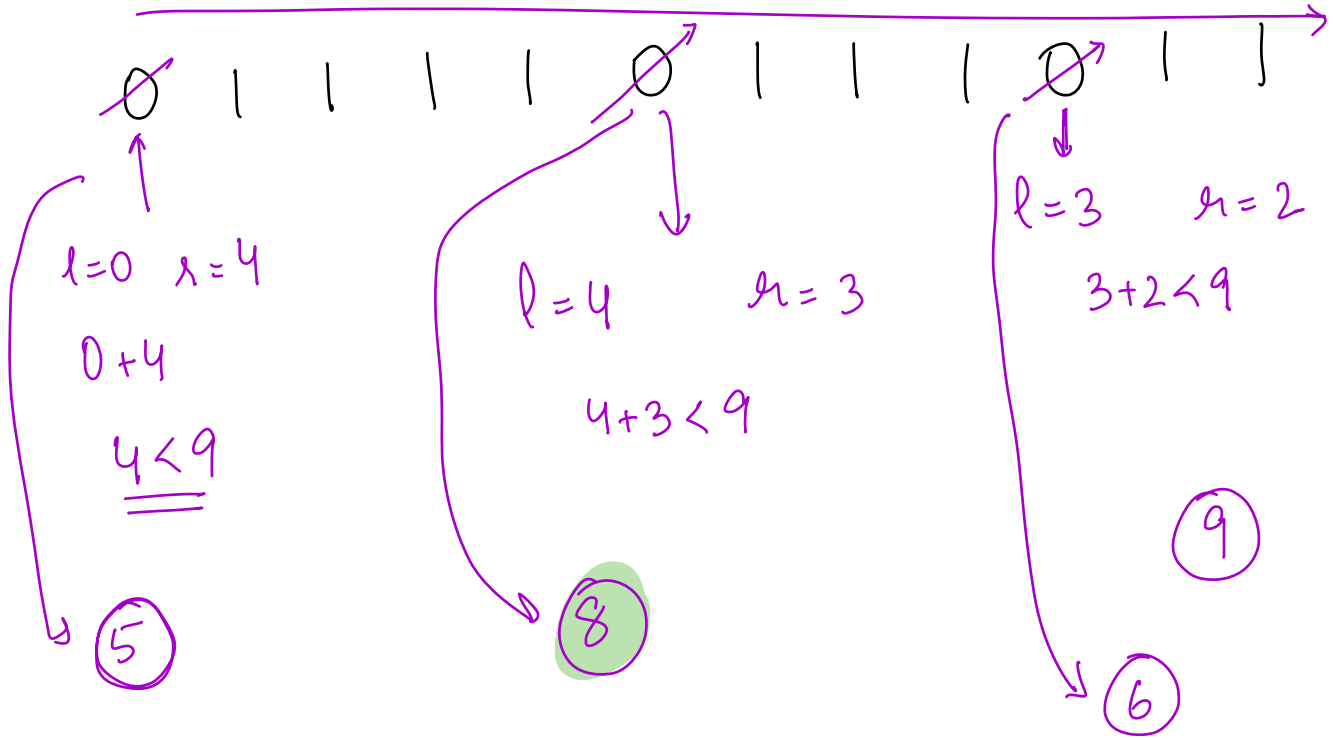
else

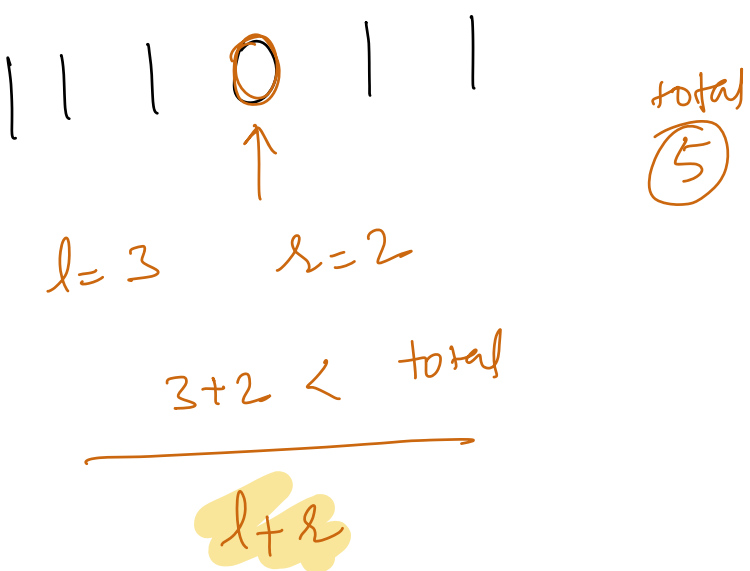
l+r.

Total count of 1s.

(6)




l=0 r=4
0+4
4<9
5
l=4 r=3
4+3<9
8
l=3 r=2
3+2<9
9
6


total
5
l=3 r=2
3+2 < total
l+r

① get one count of 1s ⇒ count_1s

```
bool flag = false
ans = 0
for(i=0; i<N; i++) {
    if (A[i]==0){
        flag = true
        // l = consecutive count of 1s on left
        l=0
        for(j=i-1; j>=0; j--){
            if(A[j]==1) l++
            else break
        }
        // r = consecutive count of 1s on right
        r=0.
        for(j=i+1; j<N; j++){
            if(A[j]==1) r++
            else break
        }
        len=l+r
        if(len<count_1s) len++
        ans = max(ans, len)
    }
}
if(flag==false) return N
return ans
```

Q. Triplets

Goldman Sachs

Given an array

Count the no. of triplets

$A[i] < A[j] < A[k]$

$i < j < k$

0	1	2	3	4	5
4	1	2	6	9	7

1 2 6

1 2 9

1 2 7

1 6 9

1 6 7

2 6 9

2 6 7

4 6 9

4 6 7

ans = 9

QUIZ

0	1	2	3	4
2	6	9	4	10

2, 6, 9

2, 6, 10

6, 9, 10

2, 4, 10

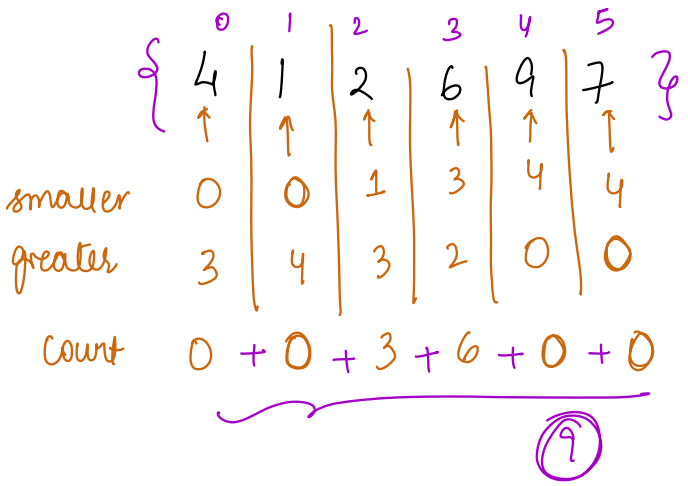
2, 9, 10

ans → 5

Brute Force

```
for(i=0; i<N; i++){
    for(j=i+1; j<N; j++){
        for(k=j+1; k<N; k++){
            if(A[i]<A[j] && A[j]<A[k])
                count ++
        }
    }
}
```

TC → O(N^3)




```
for( i = 0 ; i < N ; i++ ) {

    // smaller = count of smaller on left
    for( j = i-1 ; j >= 0 ; j-- ){
        if ( A[i] > A[j] ) smaller++
    }

    // greater = count of greater on right
    for( j = i+1 ; j < N ; j++ ){
        if ( A[i] < A[j] ) {
            greater ++
        }
    }

    count = smaller * greater
    ans = ans + count
}
```

0 1 2 3 4 5 6 7 8

1 → N

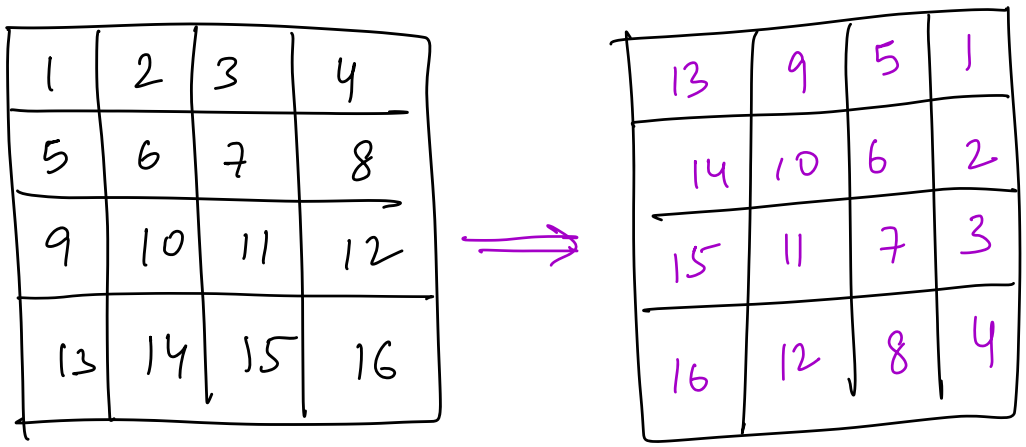
N → N + N

TC $O(N^2)$

SC $O(1)$

Rotate Matrix.

Inplace



Transpose

Compare.