Q.1 Given an array, find the max sum subarray of length k.

A =	3	9	4	-2	5	13	-7	8	k=4
	0	1	2	3	4	5	6	7	

s	e	ans
0	3	14
1	4	16
2	5	20
3	6	9
4	7	19

i) go to each subarray of length k, find its sum and the overall best is the ans.

```
int   solve (int [] A, int  k ) {

    int  s=0, e=k-1;
    int  n=A.length;
    int  ans= Integer.MIN_VALUE;

    while ( e < n )     {
            int  sum = 0;
            for (int  i = s; i<=e; i++) {
                       sum += A[i];
            }
             if (sum > ans) {
                    ans = sum;
             }
             s++;
             e++;
    }
    return  ans;
}
```

K = 3

$$A = \begin{array}{cccccc} [2 & 4 & -1 & 9 & 5 & 8] \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

s	e	sum
0	2	0 to 2
1	3	1 to 3
2	4	2 to 4
3	5	3 to 5
4	6	

TC: $O(n^2)$

SC: $O(1)$

TC: $\left(\text{No. of subarray of length } k\right) * k$

A = a b c d e f g h

A	a	b	c	d	e	f	g	h
	0	1	2	3	4	5	6	7

k	No. of subarrays	
1	8	n
2	7	$n-1$
3	6	$n-2$

$n-(k-1) \Rightarrow n-k+1$

TC: $\left(\text{No. of subarray of length } k\right) * k$

$(n-k+1) * k$

$k=1$: n

$k=n$: n

$k=\frac{n}{2}$: $\left(n-\frac{n}{2}+1\right)\frac{n}{2} \approx n^2$

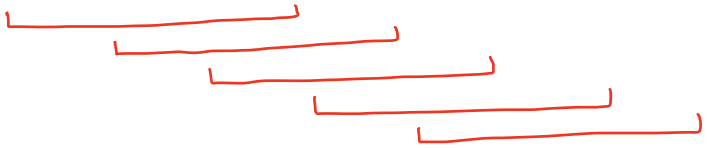
ii) prefix sum

```
int  solve (int [] A, int k) {

     int s=0, e=k-1;
     int n=A.length;
     int ans= Integer.MIN_VALUE;
     int [] ps= prefixSum(A);

     while ( e < n )     {

          int sum = 0;
          if (s= = 0) {
               sum= ps[e];
          }
          else {
               sum= ps[e] - ps[s-1];
          }

          if (sum > ans) {
               ans = sum;
          }

          s++;
          e++;
     }

     return ans;
}
```

TC: O(n)

SC: O(n)

	0	1	2	3	4	5	6	7	
A =	3	9	4	-2	5	13	-7	8	k=4



carry forward + fixed length subarray => sliding window

s	e	sum
0	3	$A[0] + A[1] + A[2] + A[3]$
1	4	$\cancel{A[0]} + A[1] + A[2] + A[3] - \cancel{A[0]} + A[4]$
2	5	$\cancel{A[1]} + A[2] + A[3] + A[4] - \cancel{A[1]} + A[5]$
3	6	$\cancel{A[2]} + A[3] + A[4] + A[5] - \cancel{A[2]} + A[6]$
4	7	$\cancel{A[3]} + A[4] + A[5] + A[6] - \cancel{A[3]} + A[7]$

$$sum - A[s-1] + A[e]$$

```
int    solve ( int[] A, int k ) {

    int n= A.length;
    int sum = 0;
    // travel the first window
    for(int i=0; i<k; i++) {
        sum+= A[i];
    }

    int ans= sum;
    int s = 1 , e = k;
    while (e<n) {
        sum= sum- A[s-1] + A[e];
        if(sum > ans) {
            ans= sum;
        }
        s++ , e++;
    }

    return ans;
}
```

k=3

$$A = \begin{array}{cccccc} [2 & 3 & -4 & 5 & 7 & 1] \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

ans= ~~1~~ ~~4~~ ~~8~~ 13

s	e	sum
		1
1	3	1 - 2 + 5 = 4
2	4	4 - 3 + 7 = 8
3	5	8 - (-4) + 1 = 13
4	6	

TC: O(n)

SC: O(1)

Q.2 Given a row and col wise sorted matrix, find if k is present in it or not.

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 10 & 20 & 30 & 40 & 50 \\ 1 & 12 & 22 & 35 & 45 & 58 \\ 2 & 18 & 25 & 49 & 54 & 68 \\ 3 & 38 & 48 & 55 & 59 & 72 \end{array}$$

$k = 49$

i) brute force : travelling entire matrix TC: $O(n*m)$

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \cancel{10} & \cancel{20} & \cancel{30} & 40 & 50 \\ 1 & \cancel{12} & \cancel{22} & \cancel{35} & 45 & \cancel{58} \\ 2 & 18 & 25 & 49 & 54 & \cancel{68} \\ 3 & 38 & 48 & 55 & \cancel{59} & \cancel{72} \end{array}$$

$k = 49$

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \cancel{10} & \cancel{20} & 30 & 40 & 50 \\ 1 & \cancel{12} & \cancel{22} & 35 & \cancel{45} & \cancel{58} \\ 2 & \cancel{18} & 25 & 49 & \cancel{54} & \cancel{68} \\ 3 & 38 & 48 & \cancel{55} & \cancel{59} & \cancel{72} \end{array}$$

$k = 38$

k = 49

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & \cancel{10} & \cancel{20} & \cancel{30} & 40 & 50 \\ 1 & \cancel{12} & \cancel{22} & \cancel{35} & 45 & \cancel{58} \\ 2 & 18 & 25 & 49 & 54 & \cancel{68} \\ 3 & 38 & 48 & 55 & \cancel{59} & \cancel{72} \end{array}$$

i	j	
0	4	
0	3	
1	3	
2	3	
2	2	→ 49

```
boolean search (int [] [] A, int k) {
    int n= A.length;
    int m= A [0]. length;
    int i= 0, j= m-1;
    while( i < n && j>=0 ) {
        if (A[i][j] == k) {
            return true;
        }
        else if (A[i][j] > k) {
            j--;
        }
        else if (A[i][j] < k) {
            i++;
        }
    }
    return false;
}
```

TC : O(n)

Dry run

```
while( i < n  &&  j >= 0 )  {
      if (A[i][j] == k)  {
              return true;
      }
      else if (A[i][j] > k) {
               j--;
      }
      else if (A[i][j] < k)  {
               i++;
      }
}
return false;
```

k = 49

A =

		0	1	j 2	3	4
	0	~~10~~	~~20~~	~~30~~	40	50
	1	~~12~~	~~22~~	~~35~~	45	58
i	2	18	25	49	54	68
	3	38	48	55	59	72

```
while( i < n  &&  j >= 0 )  {
      if (A[i][j] == k)  {
              return true;
      }
      else if (A[i][j] > k) {
               j--;
      }
      else if (A[i][j] < k)  {
               i++;
      }
}
return false;
```

k = 21

A =

		j 0	1	2	3	4
	0	~~10~~	20	30	40	50
	1	12	22	35	45	58
	2	18	25	49	54	68
i	3	38	48	55	59	72

i = 3 , j = -1

↳ out of loop

Q.3 Given a 2D matrix of N x N, print its outermost boundary in clockwise direction.

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 10 & 20 & 25 & 15 & 12 \\ 1 & 19 & 18 & 13 & 28 & 101 \\ 2 & 15 & 5 & 6 & 7 & 34 \\ 3 & 9 & 94 & 38 & 10 & 28 \\ 4 & 6 & 7 & 8 & 12 & 55 \end{array}$$

10 20 25 15 12 101 34
28 55 12 8 7 6 9
15 19

$n = 5$

$$A = \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 10 & 20 & 25 & 15 & 12 \\ 1 & 19 & 18 & 13 & 28 & 101 \\ 2 & 15 & 5 & 6 & 7 & 34 \\ 3 & 9 & 94 & 38 & 10 & 28 \\ 4 & 6 & 7 & 8 & 12 & 55 \end{array}$$

print n-1 ele L to R

print n-1 ele T to B

print n-1 ele R to L

print n-1 ele B to T

```
void   boundary (int [][] A) {
     int n= A.length;
     int i=0, j=0;
     // print n-1 values left to right
     for (int k=1; k<=n-1; k++) {
                SOP (A[i][j] + " ");
                j++;
      }

      // print n-1 values top to bottom
        for (int k=1; k<=n-1; k++) {
                SOP (A[i][j] + " ");
                i++;
        }
        // print n-1 values right to left
        for (int k=1; k<=n-1; k++) {
                SOP (A[i][j] + " ");
                j--;
        }
        // print n-1 values bottom to top
         for (int k=1; k<=n-1; k++) {
                SOP (A[i][j] + " ");
                i--;
         }
  }
```

Q.4 Given a 2D matrix of N x N, print it in spiral manner.

$$A = \begin{array}{c|cccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 10 & 12 & 94 & 55 & 18 & 6 \\ 1 & 20 & 19 & 15 & 25 & 35 & 38 \\ 2 & 41 & 42 & 49 & 54 & 48 & 55 \\ 3 & 8 & 6 & 17 & 2 & 5 & 9 \\ 4 & 13 & 2 & 3 & 40 & 78 & 3 \\ 5 & 18 & 19 & 20 & 21 & 10 & 7 \end{array}$$

i	j	n
0	0	6
1	1	4
2	2	2
3	3	0

$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 10 & 20 & 25 & 1 & 12 \\ 1 & 19 & 18 & 13 & 28 & 101 \\ 2 & 15 & 5 & 6 & 7 & 34 \\ 3 & 9 & 94 & 38 & 10 & 28 \\ 4 & 6 & 7 & 8 & 12 & 55 \end{array}$$

i	j	n
0	0	5
1	1	3
2	2	1

```
void  spiral (int [] [] A) {
    int n= A-length;
    int i=0, j=0;
    while( n>1 )  {
        // print n-1 values left to right
        for(int k=1; k<=n-1; k++) {
                sop (A[i][j]+" ");
                j++;
        }
        // print n-1 values top to bottom
        for(int k=1; k<=n-1; k++) {
                sop (A[i][j]+" ");
                i++;
        }
        // print n-1 values right to left
        for(int k=1; k<=n-1; k++) {
                sop (A[i][j]+" ");
                j--;
        }
        // print n-1 values bottom to top
        for(int k=1; k<=n-1; k++) {
                sop (A[i][j]+" ");
                i--;
        }
        i++, j++;
        n=n-2;
    }
    if(n==1) {
        sop (A[i][j]+" ");
    }
}
```

Doubts

$$\begin{array}{ccccccc} & & 6 & 9 & 8 & 10 & \\ A = & [& \cancel{1} & \cancel{2} & \cancel{1} & \cancel{4} &] \\ & & 0 & 1 & 2 & 3 & \end{array}$$

$$\begin{array}{ccccccc} & & & & & 1 & \\ & & 5 & 2 & & -2 & \\ Arr = & [& \cancel{0} & \cancel{0} & 0 & \cancel{0} &] \\ & & 0 & 1 & 2 & 3 & \end{array}$$

[5 7 7 6]

ans → add^n of A[i] + Arr[i]

(1 based)

2 3 2 → 1 2 2

1 4 5 → 0 3 5

4 4 1 → 3 3 1

s = Q[i][0] - 1;

e = Q[i][1] - 1;

val = Q[i][2];

A[s] += val;

A[e+1] += -val; (if e+1 < n)

$$\begin{array}{cccccc} & & 0 & 1 & 2 & 3 \\ A = & [& 1 & 2 & 1 & 4 \ ] \\ & & & +2 & +2 & \\ & & +5 & +5 & +5 & +5 \\ & & & & & +1 \\ \hline & & 6 & 9 & 8 & 10 \end{array}$$

Revise

↓

Assign | HW

↓

do similar ques on other platform

↳ Leetcode

↳ GFG

2, 4 → 2

0, 3 → 3

2, 4 → 5

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

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

2, 4 → 2 ✓

0, 3 → 3 ✓

2, 4 → 5 ✓

	0	1	2	3	4	5	5
0	1	1	2	2	1	0	0
1	1	1	2	2	1	0	0
2	0	0	1	1	1	0	0
3							
4							
5							
6							
⋮							
32							
	3	3	7	7	7	0	0

2 → 10
1 0

3 → 11
1 0

5 → 101
2 1 0

TC: Q + N

```
for (Q) {
    s, e, val
    for (32) → bits of val
}

for (32) {
    for (PS → 0 to N)
}
```