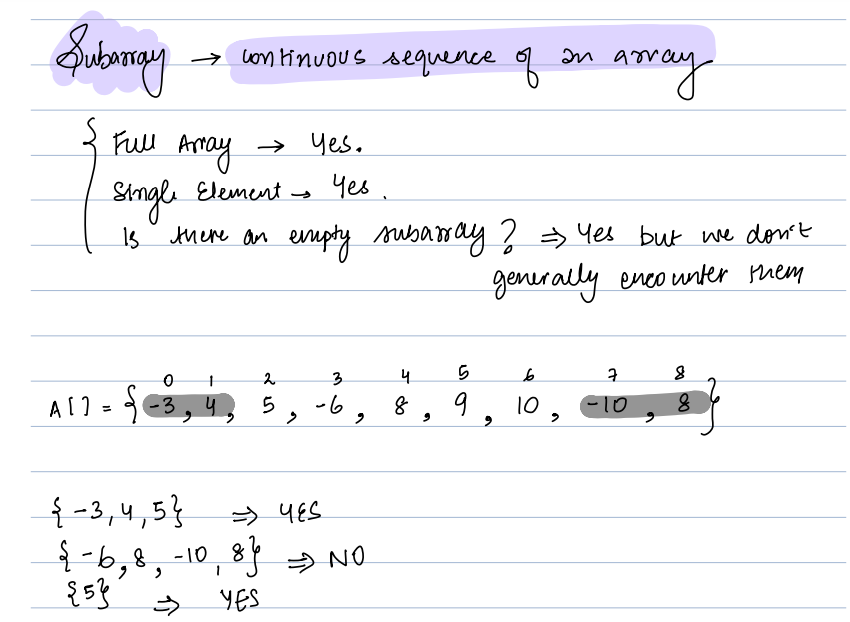
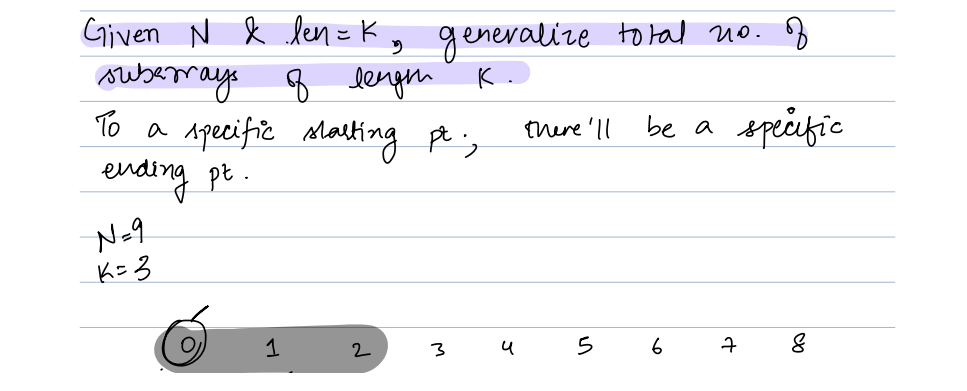
Subarrays



Total num ber of subarray: (n \* (n+1)) / 2

Q-2



BF:

Iterate for I = 0 to n-k

Iterate for j = i && j < i+k to j++

//Do nothing

Count ++;

Return Count;

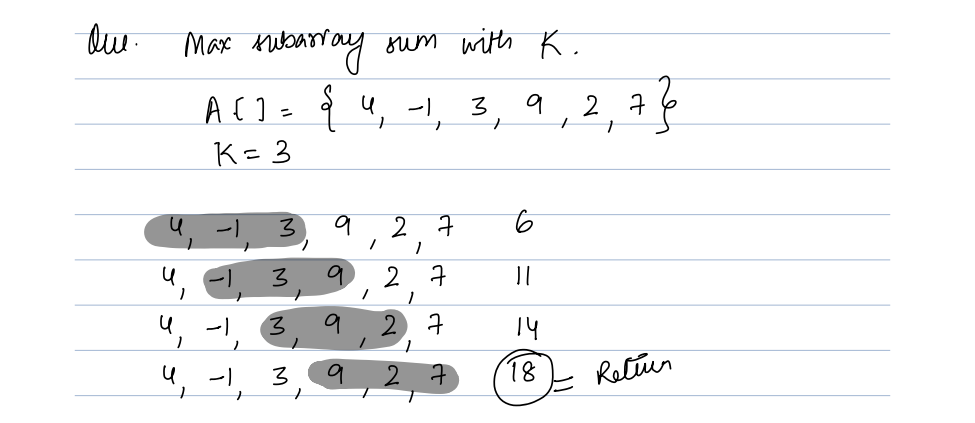
Note: Total number of subarray of length k = n –k +1

Iterate for I = 0 to n-k

S = i

E = i + k - 1

Q-3:



BF:

maxSum = int.minValue;

Iterate for I = 0 to n-k

Sum = 0;

Iterate for j = i && j < i+k to j++

S = s + A[j];

End j

maxSum = Max(S, maxSum)

End i

Return maxSum;

Optimised:

4, -1, 3, 9,2,7

Sw[] = {4, -1, 3}

Sum = sum (sw);

s = 0;

For i = k; i < n ; i++

{

Sum = sum – A[s];

Sum = sum + A[i]

S = s + 1;

maxSum = max(maxSum, sum)

}

Return maxSum;

TC = O(n)

SC = O(1)

Contribution technique

Q4:

