Note:

- **While solving the questions you must try to solve using brute force approach first.**
- **Do not consider the constraints for brute force.**
- **Once you opt to solve the question using brute force, then you should solve the problem using an optimized approach.**

1) Subarray with given sum

Given an unsorted array A of size N that contains only positive integers, find a continuous sub-array that adds to a given number S and return the left and right index(1-based indexing) of that subarray.

In case of multiple subarrays, return the subarray indexes which come first on moving from left to right.

Note:- You have to return an ArrayList consisting of two elements left and right. In case no such subarray exists, return an array consisting of element -1.

Example 1:

Input:
N = 5, S = 12
A[] = {1,2,3,7,5}
Output: 2 4
Explanation: The sum of elements
from 2nd position to 4th position
is 12.
Example 2:

Input:
N = 10, S = 15
A[] = {1,2,3,4,5,6,7,8,9,10}
Output: 1 5
Explanation: The sum of elements
from 1st position to 5th position
is 15.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
1 <= N <= 10^5
0 <= Ai <= 10^9
0<= S <= 10^9

2) Equal Left and Right Subarray Sum

Given an array A of n positive numbers. The task is to find the first index in the array such that the sum of elements before it is equal to the sum of elements after it.

Note: Array is 1-based indexed.

Example 1:

Input:
n = 5
A[] = {1,3,5,2,2}
Output: 3
Explanation: For second test case
at position 3 elements before it
(1+3) = elements after it (2+2).

Example 2:

Input:
n = 1
A[] = {1}
Output: 1
Explanation:
Since its the only element hence
it is the only point.

Expected Time Complexity: O(N)
Expected Space Complexity: O(1)

Constraints:
1 <= n <= 10^6
1 <= A[i] <= 10^8

3) Largest subarray of 0's and 1's

Given an array of 0s and 1s. Find the length of the largest subarray with equal number of 0s and 1s.

Example 1:

Input:
N = 4
A[] = {0,1,0,1}
Output: 4
Explanation: The array from index [0...3]
contains equal numbers of 0's and 1's.
Thus maximum length of subarray having
equal number of 0's and 1's is 4.
Example 2:

Input:
N = 5
A[] = {0,0,1,0,0}
Output: 2

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(N).

Constraints:
1 <= N <= 10^5
0 <= A[] <= 1

4) Subarray with 0 sum

Given an array of positive and negative numbers. Find if there is a subarray (of size at-least one) with 0 sum.

Example 1:

Input:
5
4 2 -3 1 6

Output:
Yes

Explanation:
2, -3, 1 is the subarray
with sum 0.
Example 2:

Input:
5
4 2 0 1 6

Output:
Yes

Explanation:
0 is one of the element
in the array so there exist a
subarray with sum 0.

Expected Time Complexity: O(n).
Expected Auxiliary Space: O(n).

Constraints:
1 <= n <= 10^4
-10^5 <= a[i] <= 10^5

5) Largest subarray with 0 sum

Given an array having both positive and negative integers. The task is to compute the length of the largest subarray with sum 0.

Example 1:

Input:
N = 8
A[] = {15,-2,2,-8,1,7,10,23}
Output: 5
Explanation: The largest subarray with
sum 0 will be -2 2 -8 1 7.

Expected Time Complexity: O(N).
Expected Auxiliary Space: O(N).

Constraints:
1 <= N <= 105
-1000 <= A[i] <= 1000, for each valid i

6) Smallest subarray with sum greater than x

Given an array of integers (A[]) and a number x, find the smallest subarray with sum greater than the given value. If such a subarray does not exist, return 0 in that case.

Example 1:

Input:
A[] = {1, 4, 45, 6, 0, 19}
x = 51
Output: 3
Explanation:
Minimum length subarray is
{4, 45, 6}

Example 2:
Input:
A[] = {1, 10, 5, 2, 7}
  x = 9
Output: 1
Explanation:
Minimum length subarray is {10}

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
1 ≤ N, x ≤ 10^5
0 ≤ A[] ≤ 10^4

7) Reverse sub array

Provided an array Arr of N integers, you need to reverse a subarray of that array. The range of this subarray is given by L and R.

Example 1:

Input:
N = 7
Arr[] = {1, 2, 3, 4, 5, 6, 7}
L = 2, R = 4
Output:
1 4 3 2 5 6 7
Explanation: After reversing the elements
in range 2 to 4 (2, 3, 4),
the modified array is 1, 4, 3, 2, 5, 6, 7.
Example 2:

Input:
N = 4

Arr[] = {1, 6, 7, 4}
L = 1, R = 4
Output:
4 7 6 1
Explanation: After reversing the elements
in range 1 to 4 (1, 6, 7, 4),
the modified array is 4, 7, 6, 1.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
$1 \leq N \leq 10^7$
$1 \leq Arr[i] \leq 10^6$
$1 \leq L \leq R \leq N$

8) Maximum Product Subarray

Given an array Arr[] that contains N integers (may be positive, negative or zero). Find the product of the maximum product subarray.

Example 1:

Input:
N = 5
Arr[] = {6, -3, -10, 0, 2}
Output: 180
Explanation: Subarray with maximum product
is [6, -3, -10] which gives the product as 180.
Example 2:

Input:
N = 6
Arr[] = {2, 3, 4, 5, -1, 0}
Output: 120
Explanation: Subarray with maximum product
is [2, 3, 4, 5] which gives the product as 120.

Note: Use 64-bit integer data type to avoid overflow.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
$1 \leq N \leq 500$
$-10^2 \leq Arri \leq 10^2$

9) Maximum of all subarrays of size k

Given an array arr[] of size N and an integer K. Find the maximum for each and every contiguous subarray of size K.

Example 1:

Input:
N = 9, K = 3
arr[] = 1 2 3 1 4 5 2 3 6
Output:
3 3 4 5 5 5 6
Explanation:
1st contiguous subarray = {1 2 3} Max = 3
2nd contiguous subarray = {2 3 1} Max = 3
3rd contiguous subarray = {3 1 4} Max = 4
4th contiguous subarray = {1 4 5} Max = 5
5th contiguous subarray = {4 5 2} Max = 5
6th contiguous subarray = {5 2 3} Max = 5
7th contiguous subarray = {2 3 6} Max = 6
Example 2:

Input:
N = 10, K = 4
arr[] = 8 5 10 7 9 4 15 12 90 13
Output:

10 10 10 15 15 90 90
Explanation:
1st contiguous subarray = {8 5 10 7}, Max = 10
2nd contiguous subarray = {5 10 7 9}, Max = 10
3rd contiguous subarray = {10 7 9 4}, Max = 10
4th contiguous subarray = {7 9 4 15}, Max = 15
5th contiguous subarray = {9 4 15 12},
Max = 15
6th contiguous subarray = {4 15 12 90},
Max = 90
7th contiguous subarray = {15 12 90 13},
Max = 90

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(k)

Constraints:
$1 \leq N \leq 10^5$
$1 \leq K \leq N$
$0 \leq arr[i] \leq 10^7$

10) Find Maximum Sum Strictly Increasing Subarray

Given an array of positive integers. Find the maximum sum of strictly increasing subarrays.

Example 1:

Input :
arr[] = {1, 2, 3, 2, 5, 1, 7}
Output :
8
Explanation :
Some Strictly increasing subarrays are -
{1, 2, 3} sum = 6,
{2, 5} sum = 7,
{1, 7} sum = 8,

maximum sum = 8

Example 2:

Input:
arr[] = {1, 2, 2, 4}
Output:
6

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
$1 \leq N \leq 10^5$
$1 \leq A[i] \leq 1000$