

Indore Institute of Science & Technology
Department of Computer Science & Allied Department

LIST OF Programs

SET -2

Algolution

- 1) Given an integer array `nums`, find the subarray with the largest sum, and return its sum.
Input: `nums = [-2,1,-3,4,-1,2,1,-5,4]`
Output: 6
Explanation: The subarray `[4,-1,2,1]` has the largest sum 6.

- 2) Given an integer array `arr[]` of size `n`, the task is to find the count inversions of the given array.
Two array elements `arr[i]` and `arr[j]` form an inversion if `arr[i] > arr[j]` and `i < j`.
Input: `arr[] = {7, 2, 6, 3}`
Output: 4
Explanation: Given array has 4 inversions: `(7, 2), (7, 6), (7, 3), (6, 3)`

- 3) Given an integer array `nums` and an integer `k`, return the `k`th largest element in the array.
Note that it is the `k`th largest element in the sorted order, not the `k`th distinct element.
Solve it without sorting.
Example:
Input: `nums = [3,2,1,5,6,4], k = 2`
Output: 5

- 4) A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward.
Alphanumeric characters include letters and numbers.

Given a string `s`, return `true` if it is a palindrome, or `false` otherwise.
Example 1:
Input: `s = "A man, a plan, a canal: Panama"`
Output: `true`
Explanation: `"amanaplanacanalpanama"` is a palindrome.
Example 2:
Input: `s = "race a car"`;
Output: `false`

- 5) Given an array `arr[]` of size `n`, return an equilibrium index (if any) or -1 if no equilibrium index exists. The equilibrium index of an array is an index such that the sum of elements at lower indexes equals the sum of elements at higher indexes.
Note: Return equilibrium point in 1-based indexing. Return -1 if no such point exists.
Examples:

Input: arr[] = {-7, 1, 5, 2, -4, 3, 0}

Output: 4

Explanation: In 1-based indexing, 4 is an equilibrium index, because:

$$\text{arr}[1] + \text{arr}[2] + \text{arr}[3] = \text{arr}[5] + \text{arr}[6] + \text{arr}[7]$$

Input: arr[] = {1, 2, 3}

Output: -1

Explanation: There is no equilibrium index in the array.

Input: arr = {1, 3, 5, 2, 2}

Expected Output: 2

Explanation: At index 2, the sum of elements on the left ($1 + 3 = 4$) is equal to the sum of elements on the right ($2 + 2 = 4$).

6) Given a sorted array A (sorted in ascending order), having N integers, find if there exists any pair of elements ($A[i]$, $A[j]$) such that their sum is equal to X.

a) Using Naive Approach

b) Using Two Pointer Technique

Input: A = [1, 2, 4, 5, 7, 11]

N = 6

X = 9

Expected Output: Yes

(Explanation: The pair (2, 7) sums to 9.)

8) Given an array of integers of size 'n', Our aim is to calculate the maximum sum of 'k' consecutive elements in the array (Using Sliding Window Technique)

Input : arr[] = {100, 200, 300, 400}, k = 2

Output : 700

Input : arr[] = {1, 4, 2, 10, 23, 3, 1, 0, 20}, k = 4

Output : 39

We get maximum sum by adding subarray {4, 2, 10, 23} of size 4.

Input : arr[] = {2, 3}, k = 3

Output : Invalid

There is no subarray of size 3 as size of whole array is 2.

9) You are given an array of prices where prices[i] is the price of a given stock on an ith day. You want to maximize your profit by choosing a single day to buy one stock and choosing a different day in the future to sell that stock. Return the maximum profit you can achieve from this transaction. If you cannot achieve any profit, return 0

Input: prices = [7, 1, 5, 3, 6, 4]

Output: 5

Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6), profit = $6 - 1 = 5$.

10) Given an integer array, find the maximum length subarray having a given sum. (USE HASHMAP Concept)

Input: nums[] = { 5, 6, -5, 5, 3, 5, 3, -2, 0 } target = 8

Output: { -5, 5, 3, 5 }

Subarrays with sum 8 are { -5, 5, 3, 5 }, { 3, 5 }, { 5, 3 }

The longest subarray is { -5, 5, 3, 5 } having length 4.
