



Course Title: Data Structure & Algorithm Lab II

Course Code: CSE2218

Trimester & Year: Spring 2022

Section: E

Credit Hours: 1.0

AZ

ASSIGNMENT 01: Divide and Conquer

Q1: Maximum Subarray Implementation

Given an integer array `nums`, find the contiguous subarray (containing at least one number) which has the largest sum and return *its* **sum**.

A **subarray** is a **contiguous** part of an array.

Example 1:

Input: `nums = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`

Output: 6

Explanation: `[4, -1, 2, 1]` has the largest sum = 6.

Example 2:

Input: `nums = [1]`

Output: 1

Example 3:

Input: `nums = [5, 4, -1, 7, 8]`

Output: 23

Constraints:

- `1 <= nums.length <= 105`
- `-104 <= nums[i] <= 104`



UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)

Q2: The Perfectionist

Alif is a perfectionist about numbers. He loves to arrange things in order and sticks to his “Golden rule” that every set of numbers must be in ascending order. Unfortunately, that is not always the case. Alif defines that when a smaller number comes after a larger number in the set then number of violations are required to fix the order.

Given a set of integers, Alif needs to find out the total number of such violations.

Input:

- The first line contains **n**, the number of integers.
- The second line contains n space separated integers **a₀ ... a_{n-1}**

Output: The output is an integer indicating the total number of violations.

Bonus: Time Complexity should be less than or equals to **$O(n \log n)$**

Example:

Sample Input	Sample Output	Explanation
5 4 5 6 7 1	4	1 violates which requires 4 violation to be fixed as 1 4 5 6 7.
5 5 4 3 2 1	10	4 violates which requires 1 violation to be fixed as 4 5 3 2 1. Then again 3 violates which requires 2 violation to be fixed as 3 4 5 2 1. Then again 2 violates which requires 3 violation to be fixed as 2 3 4 5 1. Then again 1 violates which requires 4 violation to be fixed as 1 2 3 4 5. Total violations required = 1+2+3+4 =10



Q3: Kth Largest Element in an Array

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.

Example 1:

Input: `nums = [3, 2, 1, 5, 6, 4]`, `k = 2`

Output: 5

Example 2:

Input: `nums = [3, 2, 3, 1, 2, 4, 5, 5, 6]`, `k = 4`

Output: 4

Constraints:

- `1 <= k <= nums.length <= 104`
- `-104 <= nums[i] <= 104`



Q4: Reverse Pairs

Given an integer array `nums`, return *the number of **reverse pairs** in the array*.

A reverse pair is a pair (i, j) where $0 \leq i < j < \text{nums.length}$ and $\text{nums}[i] > 2 * \text{nums}[j]$.

Example 1:

Input: `nums = [1,3,2,3,1]`

Output: 2

Example 2:

Input: `nums = [2,4,3,5,1]`

Output: 3

Constraints:

- $1 \leq \text{nums.length} \leq 5 * 10^4$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$