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

Course Title: Data Structure & Algorithm Lab II **Course Code: CSE2218**

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.

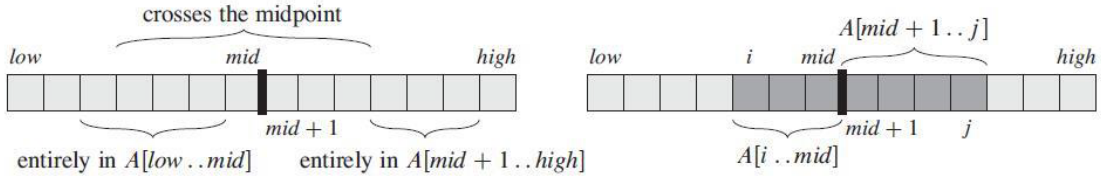
Example:

```
Input: nums = [-2, 1,-3, 4,-1, 2, 1,-5, 4]
Output: 6
Explanation: [4,-1, 2, 1] has the largest sum = 6.
```

Steps to follow:

Each time divide the array into two halves.

a. Recursive call the first half to return you the maximum subarray sum of that portion.
b. Recursive call the second half to return you the maximum subarray sum of that portion.
c. Maximum subarray may exist around the mid-point. So, calculate the maximum subarray sum across the split boundary.



How to calculate the maximum crossing subarray sum:

```
FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
1  left-sum = −∞
2  sum = 0
3  for i = mid downto low
4      sum = sum + A[i]
5      if sum > left-sum
6          left-sum = sum
7          max-left = i
8  right-sum = −∞
9  sum = 0
10 for j = mid + 1 to high
11     sum = sum + A[j]
12     if sum > right-sum
13         right-sum = sum
14         max-right = j
15 return (max-left, max-right, left-sum + right-sum)
```

ii. Base condition: If the array contains only 1 item then the maximum subarray sum is the item itself.

Q2: Quick Sort Implementation

Given an integer array `nums`. Sort the array in descending order using Quicksort algorithm and print the array.

Sample Input	Sample Output
6 4 5 6 7 1 3	7 6 5 4 3 1