UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)
CSE 2218: Data Structures and Algorithms Laboratory
Assignment 1

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.

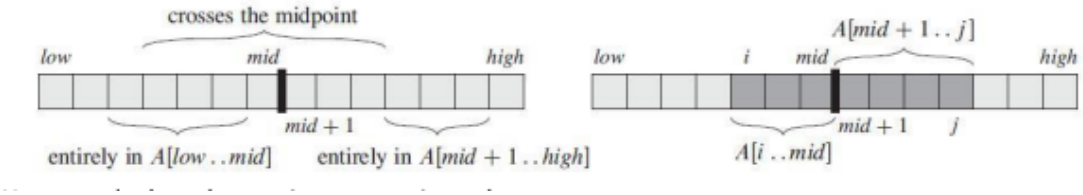
Example 1:
Input: nums = [-2, 1,-3, 4,-1, 2, 1,-5, 4]
Output: 6
Explanation: [4,-1, 2, 1] has the largest sum = 6.

Example 2:
Input: nums = [1]
Output: 1

Example 3:
Input: nums = [5, 4,-1, 7, 8]
Output: 23

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



Calculating 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)
```

Q2: Count Set-bits of number using Recursion

For a given number N, the task is to find the number of set bits in its binary representation using recursion.

Task Format:

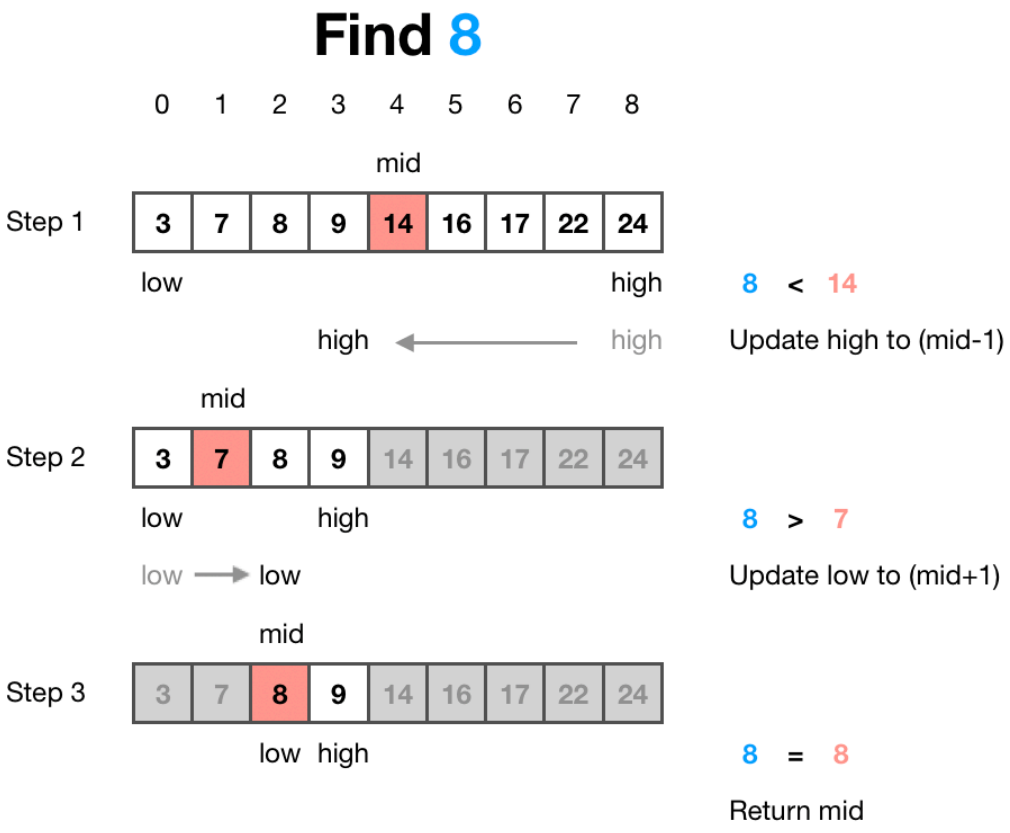
Input 21	Output 3

Given Input n=21
The binary representation of 21 is 10101. The total number of 1 in the string is 3. [Hint: Keep track of the number of 1's while converting the number from decimal to binary]

Q3: Recursive Implementation of Binary Search

Binary Search Algorithm is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(log N).

The binary search works by recursively dividing a sorted array and finding an index in the sorted array. The step by step visualization of binary search is shown below:



If the element that you want to search is lesser than the mid value, for a sorted array, the element must be in the range of subarray smaller than the mid value. In step 2, the search space is then updated to only 3,7,8,9 instead of the entire array. We apply this strategy until the last element we find is our expected index.

The pseudocode for binary search is shown below:

```
int RBinSearch(int a[], int l, int h, int key)
{
        int mid;
        if (l <= h)
        {
                mid = (l + h) / 2;
                if (key == a[mid])
                        return mid;
                else if (key < a[mid])
                        return RBinSearch(a, l, mid - 1, key);
                else
                        return RBinSearch(a, mid + 1, h, key);
        }
        return -1;
}
```

Submission Guidelines:

- Implement the following tasks using your preferred programming language(C/C++)
- Compile your answers into a single document
- Show a step by step explanation of your implementation along with your expected output.
- Mention your name and ID on the cover page of your assignment.

Some Helpful Tutorials:

- https://www.youtube.com/watch?v=ohHWQf1HDfU
- https://youtu.be/OKCBidfgj3Q?si=_hU8vHOmcML9j4-I
- https://www.youtube.com/watch?v=-bQ4UzUmWe8

Best of luck to all my students!