Lab - 4

Evaluative Questions

1. **(5 points) 2-D Array, Recursion: CO1**

Imagine you are an explorer lost in a maze. You can only move in four directions: up, down, left, and right. Write a recursive function to help you find your way out of the maze. The maze is represented as a 2D grid where '0' represents an open path and '1' represents a wall. Your function should return true if you can escape the maze, and false otherwise.

Example:
Maze :
0 0 0
1 1 0
0 0 0

Expected Output: true
Maze :
0 0 1
1 1 1
0 0 1

Expected Output: false

2. **(5 points) Array, Recursion: CO 1, CO2**

Write a divide-and-conquer algorithm that recursively divides the array into smaller subarrays, computes the maximum subarray sum for each subarray, and combines the results to reveal the overall maximum subarray sum. What is the time complexity of your algorithm? Mention as comments.

Scenario 1:
Array: [-2, 1, -3, 4, -1, 2, 1, -5, 4]
Expected Output: 6
Explanation: The contiguous subarray [4, -1, 2, 1] has the largest sum of 6.

Scenario 2:
Array: [1, 2, 3, 4, -10, 6, 7, 8, 9]
Expected Output: 30
Explanation: The entire array [1, 2, 3, 4, -10, 6, 7, 8, 9] has the largest sum of 30.

Scenario 3:
Array: [-5, -2, -6, -1, -2, -3, -7, -4]
Expected Output: -1
Explanation: The contiguous subarray [-1] has the largest sum of -1.

Practice Questions

1.

You are given a certain number of sapphires (N). You want to create a triangular sculpture using these sapphires, where each row of the sculpture has one more sapphire than the row above it. Write a C++ program to find the maximum height of the sculpture you can create using the given number of sapphires. Ensure that your program has a worst-case time complexity of O(log(N)).

Test cases:

With N = 5 sapphires,we can create a triangular sculpture with a maximum height of 2 rows. The first row contains 1 sapphire, and the second row contains 2 sapphires, totaling 3 sapphires, which is less than the available 5 sapphires. Therefore, the maximum possible height is 2 rows. The sculpture would appear as follows

*

**

**

2. A skilled adventurer, the Limited Turns Coin Collector, embarks on a quest to collect as much wealth as possible. However, due to the magical nature of the path, the collector can only take a limited number of turns, denoted by k.

   At the beginning of each turn:

   - The collector stands at the entrance of the path, facing a row of coins.
   - The collector can choose to pick either the first or the last coin in the row.

- After each pick, the remaining coins (after removing the first and the last coins of previous row) form a new row.

The collector's goal is to maximize the total value of coins collected within the k turns.

Example:
Input: coins = [1,2,3,4,5,6,1], k = 3
Output: 12
Explanation: After the first step, your score will always be 1. However, choosing the rightmost coin first will maximize your total score. The optimal strategy is to take the three cards on the right, giving a final score of 1 + 6 + 5 = 12.