United International University

Department of Computer Science and Engineering

Course Code: CSI 217 | Course name: Data Structure and Algorithms - I Laboratory Section E

Summer 2023 # Final Class Performance

Total Marks: 25 **Deadline: 27 - 08 -2023**

Question 1 : Special Stack (Mark 9)

In a mystical realm where algorithms come to life, you've stumbled upon an extraordinary data structure known as the "Enchanted Stack." This stack isn't bound by the ordinary rules of stacks; it possesses two magical operations: popEnd() and popMagic().

push(val): Add an element val to the top of the stack.
popEnd(): Remove and return the last element from the stack.
popMagic(): Remove and return the **middle element** from the stack. In case the number of elements is even, choose the second middle element.

Question 2 : Binary Search Tree (Mark 8)

Finding Sum of Two Nodes : You are allowed to use the Binary Search Tree (BST) implementation that supports the find(number) function. Using this function, your task is to implement another function called **isSumPossible(root, sum).**

The isSumPossible function takes as input the root node of a BST and a target sum. Your goal is to determine whether there exist two distinct nodes in the BST such that their values add up to the given sum.

Here's the Example :

For the BST above, isSumPossible(root, 17) should return True since 7 + 10 = 17.
For the same BST, isSumPossible(root, 25) should return True since 5 + 20 = 25.
For the same BST, isSumPossible(root, 12) should return False as no two distinct nodes add up to 12.

Question 3 : The Maze of Mysteries - Finding Paths with Treasures with certain moves (Mark 8)

You've now mastered the art of maze navigation and treasure hunting in offline assignments ! In this challenge, you're tasked with determining **whether you can find a treasure in just 3 moves**.

You need to decide whether you can reach a treasure in exactly 3 moves from your starting position, while following the rules of the maze (avoiding walls).
If you can reach a treasure in 3 moves, print **"Eureka"**.
If there's no way to reach a treasure in 3 moves, print **"Treasure hunt is boring".**

Here's the Example :

Consider the following maze:

```
maze = [
   [0, 1, 0, 0, 2],
   [0, 0, 0, 1, 0],
   [0, 3, 1, 0, 0],
   [0, 1, 0, 1, 0],
   [0, 0, 0, 0, 0]
]
```

For this maze and starting position (1, 0), you should print **"Eureka and Can reach End"**

```
maze = [
   [0, 1, 0, 0, 2],
   [0, 0, 0, 1, 0],
   [0, 0, 1, 0, 0],
   [0, 1, 0, 1, 0],
   [0, 0, 0, 0, 3]
]
```

For this maze and starting position (1, 0),You should print **"Treasure hunt boring and Can reach End"** because there's no path to reach a treasure in exactly 3 moves.