Data Structures and Algorithms
CS-F22
LAB-05

Issue Date: July 26,2024

Start Time: 8:15 AM
Completion Time: 11:15 M

Total Marks : 60

Objective:

In this lab, students will understand about working of stack, queue and sorting of arrays. The objective is to reinforce the concepts studied so far.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Your laptops must be on airplane mode.
3) Anyone caught in an act of plagiarism would be awarded an "F" grade in this Lab.

TASK 01: Farthest Building [20 Marks]

Given an array of integers representing the heights of buildings, along with a certain number of bricks and ladders.

Your task is to determine the farthest building you can reach if you use the given ladders and bricks in the most efficient way.

You start at building 0 and can move to the next building by using either bricks or ladders as needed.

- You only require to use ladders and bricks when current building height is lower than that of next building.
- If the height of the current building is less than the height of the next building, you can choose to use bricks (equal to the height difference) or one ladder (ladders can reach any height).

Function prototype:

int farthestBuilding(int * heights, int bricks, int ladders)

Sample run:

Input: heights = [4,2,7,6,9,14,12], bricks = 5, ladders = 1

⇨ Go to building 1 without using anything.
⇨ Go to building 2 using 5 bricks.
⇨ Go to building 3 without using anything.
⇨ Go to building 4 using your ladder.

Output: 4

TASK 02: Operations to sort the array [25 Marks]

Given an integer array nums. Return the **minimum number** of operations to make an array that is sorted in non-decreasing order.

- In one operation you can replace any element of the array with any two elements that sum to it.

For example, consider nums = [2,4,8]. In one operation, we can replace 4 with 2 and 2 and convert nums to [2,2,2,8].
Function prototype:

int minOperations(int * nums, int n)

Sample run:
Input: [6,18,6]

[6,18,6] ➔ [6,6,12,6] (18 replaced with 6 and 12)
[6,6,12,6] ➔ [6,6,6,6,6] (12 replaced with 6 and 6)

Output: 2

Note: *Provide a solution with linear time complexity O(n) → (Use single loop) to get maximum points.*

TASK 03: Clumsy Factorial [15 Marks]

Write a function to find the **clumsy factorial** of the given integer.
A clumsy factorial is found by using the integers in decreasing order and applying the operations multiply '*', divide '/', add '+', and subtract '-' in a repeated cycle.
Function prototype:

int clumsyFactorial(int n)

Note: *These operations are applied using the usual order of operations of arithmetic.*

Sample run:
Input: 10
Output: 12 ➔ 10 * 9 / 8 + 7 - 6 * 5 / 4 + 3 - 2 * 1