Assignment 1 - Data Structures I

Due: **September 25 2022 11:59PM**

1 TA

Any Questions Regarding the Assignment: Yulong Fan (fan176@pnw.edu)

2 Arrays [10 Points]

Given an array that may contain duplicate integers, return an array without duplicates and its size.

- Test case 1: Input: array = 4, 1, 1, 10, 10, 5, array.length = 6 Output: array = 4, 1, 10, 5, array.length = 4
- Test case 2: Input: array = 5, 1, 10, 0, 1, 5, 4, array.length = 7 Output: array = 5, 1, 10, 0, 4, array.length = 5
- Test case 3: Input: array = 1, 2, array.length = 2 Output: array = 1, 2, array.length = 2

3 Stacks[10 Points]

Given a string consist of brackets such as: , , (,) , [,]. Use stack to determine rather the brackets are balanced.

- Test case 1: Input: string = "[()]" Output: [()] is balanced
- Test case 2: Input: string = "()[]" Output: ()[] is balanced
- Test case 3: Input: string = "[(][" Output: [(][is not balanced

4 Queues[20 Points]

You have a RecentCounter class which counts the number of recent requests within a certain time frame.

Implement the RecentCounter class:

RecentCounter() Initializes the counter with zero recent requests. int ping(int t) Adds a new request at time t, where t represents some time in milliseconds, and returns the number of requests that has happened in the past 3000 milliseconds (including the new request). Specifically, return the number of requests that have happened in the inclusive range [t - 3000, t]. It is guaranteed that every call to ping uses a strictly larger value of t than the previous call.

Input ["RecentCounter", "ping", "ping", "ping", "ping"] [[], [1], [100], [3001], [3002]] Output [null, 1, 2, 3, 3]

Explanation RecentCounter recentCounter = new RecentCounter();

recentCounter.ping(1);

// requests = [1], range is [-2999,1], return 1 recentCounter.ping(100); // requests = [1, 100], range is [-2900,100], return 2 recentCounter.ping(3001); // requests = [1, 100, 3001], range is [1,3001], return 3 recentCounter.ping(3002); // requests = [1, 100, 3001, 3002], range is [2,3002], return 3

5 Linked List[10 Points]

Delete the middle element of a Linked List (using two pointer method).

6 Instructions

- Please comment your code properly for clarity.
- Please put all java files in a zip folder during submission.
- The TA can run more test cases to verify your program.
- 1 bonus point (each) will be given if you can identify the correct time complexity.