Virginia Commonwealth University, Computer Science

CMSC 401: Algorithm Analysis with Advanced Data Structures

Assignment 2

February 10, 2020

Due Feb 24th, 2020 at 11:59pm by uploading your solutions to Blackboard. In this assignment, you are expected to work individually. You may use any sources that you want, but you must cite them. The submission should be a zip file named "A2_$< your name >$.zip", which contains all the source code as required. You should write your program following Google code style (https://google.github.io/styleguide/javaguide.html). My office hours are M/T 1-2pm.

1 Write a program as required below. (40 pts)

You are asked to implement a quicksort program named MyQuicksort that satisfies the following requirements:

1. Your sorting algorithm must be quicksort.
2. Your quicksort algorithm will always pick the right most element (no random swap) to be the pivot and use the divide and conquer methodology to sort the array.
3. Your program will take a sequence of integers separated by spaces from console as the input array.
4. It will print the sorted array numbers separated by spaces.
5. It will print the number of comparisons executed in the program.
6. For example

   ```
   >java MyQuicksort 5 10 7 101
   ```

```
5 7 10 101
6
```

2 Write a program as required below. (60 pts)

You are asked to implement a binary search tree class named MyBinaryTree that satisfies the following requirements:

1. It will use the following class as the binary tree node.

   ```
   class BTNode {
   int key;
   BTNode parent;
   BTNode left;
   BTNode right;
   }
   ```

2. It should contain but not limited to the following methods:

   void add(int newNum) // This method will add a new integer number into the binary search tree.

   boolean delete(int delNum) // Delete the number *delNum* if it exists. Return true if succeeds and or return false if the number does not exist.

   boolean contains (int num) // Check whether the tree contains the number *num* or not.

   String inOrder() // Return the inorder traversal result (numbers are separated by spaces)

   String preOrder() // Return the preorder traversal result (numbers are separated by spaces)

   String postOrder() // Return the postorder traversal result (numbers are separated by spaces)

3. It should have a main method that reads a sequence of numbers from the terminal and print the sorted results. For example:

   ```
   >java MyBinaryTree 5 10 7 101 42 1 21 9998 -65
   -65 1 5 7 10 21 42 101 9998
   ```