Version: 20-Apr-2022 5PM

# CS 2223 D21 Term. Homework 4

## Homework Instructions

- This homework is to be completed individually. If you have any questions as to what constitutes improper behavior, review the examples as I have posted online http://web.cs.wpi.edu/~heineman/html/teaching_/cs2223/d22/#policies.
- Due Date for this assignment is **6PM Monday May 2nd** which is the day before the end of the term.
- Submit your assignments electronically using the canvas site for CS2223. Login to canvas.wpi.edu and locate HW4. You must submit a single ZIP file that contains all of your code as well as the written answers to the assignment.
- All of your Java classes must be defined in a packager USERID where USERID is your CCC user id.
- Submission information is found at the end of this document.

## Primary Instructions

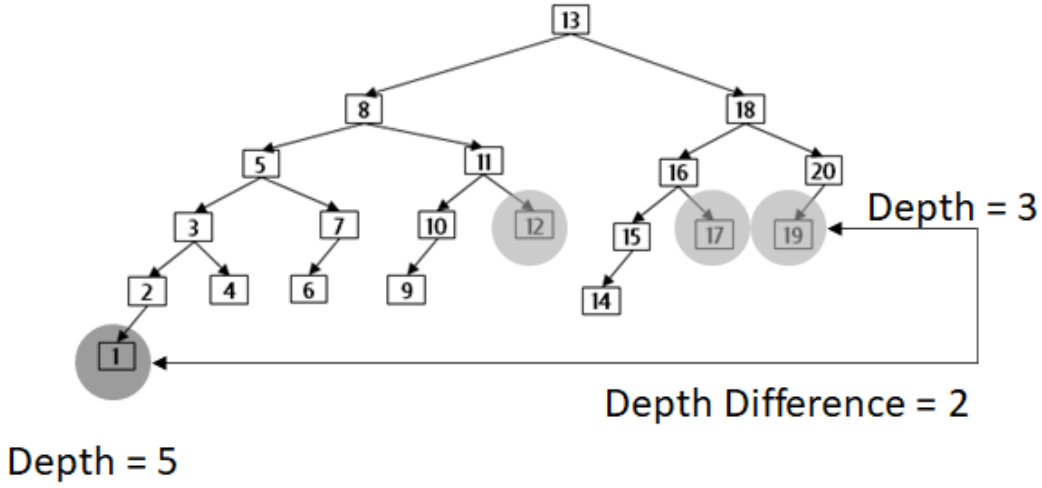
This homework assignment is about problem solving. Using the data structures and algorithms presented over the past few weeks (including this week), this assignment gives you the opportunity to take the ideas presented in class and solve problems.

Version: 20-Apr-2022 5PM

## Q1. AVL Trees (40 pts)

Each node in an AVL tree satisfies the **AVL Property**, namely that the height difference for any node is -1, 0 or +1. Given a random AVL tree, how many of its nodes have a height difference of zero?

I have another question, exemplified by the following valid AVL tree:




There are three leaf nodes whose depth is 3. In fact, **this is the minimum depth of any leaf node**. There is one node whose depth is 5. In fact, **this is the maximum depth of any leaf node**. Recall from lecture that the height of a binary tree is the maximum number of edges to reach any leaf node. So, if you know the height of a binary tree, you know the maximum depth of a leaf node in the tree.

Given an AVL tree, **what is its depth difference** which is defined as "(**maximum depth of any leaf node**) – (**minimum depth of any leaf node**)"? In the above example with 20 nodes, the depth difference is 2.

Copy the `algs.hw4.Counting` class into your **`USERID.hw4`** package. Also copy the `algs.hw4.AVL` and `algs.hw4.RedBlackBST` classes since you will be modifying these classes. In particular, **for both of these classes** you need to complete the implementations for the following four methods:

```
/** Return a Symbol Table (in this case, SeparateChainingHashST) which records the
 * counts of nodes in the tree that have a height differential of -1, 0 or 1.
 * This means there will be just three keys in this symbol table, and the sum of the
 * VALUES associated with these keys will be N. */
public SeparateChainingHashST<Integer,Integer> counts() { … }

/** Helper method that seeks to update the given symbol table, ht, with the height
 * differentials in the sub-tree rooted at parent.*/
private void counts(Node parent, SeparateChainingHashST<Integer,Integer> ht) { … }

/** Public facing API call that returns the minimum depth of a leaf node in the
 * tree. Recall that the depth of the root node is 0 (and similarly, the depth of an
 * empty tree is -1). The depth of a leaf node is the number of edges required to
 * traverse to that node from the root. */
public int minDepth() { … }
/** Helper method that seeks to find the leaf with minimum depth from the root in the
```

```
 * sub-tree rooted at parent, assuming that 'parent' is at depth currentDepth
 * and that the minimum depth so far recorded for a leaf node is lowestSoFar. */
private int minDepth(Node parent, int currentDepth, int lowestSoFar) { … }
```

For RedBlackBST, you will additionally need to implement the following:

```
/** Helper method to determine height differential for a given node by subtracting
 * leftH - rightH. Note that for a RedBlackBST tree, the height differential is NOT
 * restricted to just -1, 0 or 1. */
 private int heightDifference(Node node) { … }
```

For N varying from one less than a power of 2, namely from 31 to 262143, the Counting class runs 100 independent trials. For each trial, it constructs an array containing the integers from 1 to N-1 and randomly shuffles the array. You will see that it first calls `StdRandom.setSeed(T)` where T is the trial number (from 0 to 99) to ensure that these results can be replicated. In fact, the table you output should be identical to mine (although I only show the first two rows in this homework description).

For each trial, construct a new USERID.hw4.AVL tree and a USERID.hw4.RedBlackBST tree. Then insert the values (from left to right) from the shuffled array into the AVL and RedBlackBST trees. For the resulting AVL and RedBlackBST trees, you are to (a) count the nodes whose height differential is zero; and (b) compute the depth difference. Across all 100 trials, you are to (a) record the maximum height of the tree over all trials; (b) record the **max count** of nodes with height differential of 0 over all trials; and (c) record the **max depth difference** you computed over all trials.

When done generate a table that looks like the following. I give you the first two rows (which you should be able to match exactly). Be sure to produce table up to N = 262143.

```
N       MaxAVHt MaxAVDf MaxAVZr AVZero% MaxRBHt MaxRBDf MaxRBZr RBZero%
31      5       2       25      80.645  6       3       24      77.419
63      7       3       51      80.952  8       4       48      76.190
…
```

The columns are identified as follows:

- MaxAVHt records the maximum height of any of the 100 random AVL trees of size N
- MaxAVDf records the maximum depth difference over all 100 random AVL trees of size N
- MaxAVZr records the maximal count of nodes in a single AVL tree whose height difference is 0
- AVZero% simply reports the percentage of (MaxAVZr/N)
- MaxRBHt records the maximum height of any of the 100 random RedBlack trees of size N
- MaxRBDf records the maximum depth difference over all 100 random RedBlack trees of size N
- MaxAVZr records the maximal count of nodes in a single RedBlack tree whose height differential is 0
- RBZero% simply reports the percentage of (MaxRBZr/N)

## Q1. Bonus (1pt)

Can you sample the height differentials for RedBlack Tree (which are not restricted to -1, 0, and 1), with increasing N, and speak to the maximum values found, plus the frequency of the counts? Is it uniform?

Version: 20-Apr-2022 5PM

## Q2 Learn to work with graphs and Breadth First Search (60 total pts)

On Day 22, I showed how to work with highway maps as graphs. In the example I showed, I arbitrarily picked two vertices to conduct the search. For this assignment, you are to work with Breadth First Search to compute specific paths through the highway graph. Feel free to use the ideas from **`algs.days.day22.BreadthFirstPaths`**.

Copy the algs.hw4.MapSearch class into USERID.hw4 and modify it to complete this question.

## Q2.1 Standard Paths (25 pts)

There are two scenarios to tackle:

- Using Breadth First Search, compute the shortest path (in terms of total number of highway segments) from the western-most highway location to the eastern most highway location. Complete the implementation of **`westernMostVertex()`** and **`easternMostVertex()`**. Be sure to identify the label associated with each vertex and print the total number of edges in the path.
- Using Breadth First Search, compute the shortest path (in terms of total number of highway segments) from the southern-most highway location to the northern-most highway location. Complete the implementation of **`southernMostVertex()`** and **`northernMostVertex()`**. Be sure to identify the label associated with each vertex and print the total number of edges in the path.

Given the London Highway system, you will find that the following labels are associated with these different most distance edge points. The following are the actual labels in some random order (if you are curious, you can find the raw map data in `algs.hw4.resources`) . Only you will be able to know which ones are West, East, South or North once you complete your changes.

- A308/B3028
- B2036/B2037
- B197@StaRd
- A13@+X674120

## Q2.2 Demonstrate why Depth First Search is inappropriate here (10 pts)

For the same two considered scenarios above, now complete a Depth First Search and report the total number of edges for each one, as you did with Breadth First Search.

## Q2.3 Eliminate M25 highway from consideration (25 pts)

In some GPS applications, the user can request to avoid specific highways roads. For this question, you are to complete the implementation of **`Information remove_M25_segments(Information info)`** which is in the `MapSearch` class. This method will return a new **`Information`** object containing *a new graph* for the London highway segments except those include the dreaded M25. Note that you cannot currently remove an edge from a **`Graph`** object, so instead you will create a new graph using its original vertices but you will only add *edges that do not involve any reference to the M25 highway*. Find instructions in the `MapSearch` class.

Now go back and compute the above standard paths for Breadth First Search only. Describe the impact of not being able to use the M25. Describe the change (if there is one) on both scenarios (namely West-to-East and North-to-South), specifically regarding the total number of edges.

Solution should look something like this:

```
BreadthFirst Search from West to East:
BFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.

BreadthFirst Search from South to North:
BFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.

DepthFirst Search from West to East:
DFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.

DepthFirst Search from South to North:
DFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.

Now without M25 edges...

WEST to EAST
BFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.

NORTH to SOUTH
BFS: SOMEPLACE (999) to SOMEPLACE (999) has 999 edges.
```

## Q2.4 Bonus (1 pt)

In the **`algs.hw4.map.GPS`** class there is a method that computes the distance in miles between any two GPS coordinates. Use this method to compute the length in mileage for the shortest computed paths for the two pairs of edge points.

Determine if the total mileage is longer (or shorter) when excluding M25 from consideration.

Consider using/modifying the **`BFSMapAnimation`** example from day 22 to generate images.

## Q2.5 Bonus (1 pt)

The M25 is considered an orbital highway around London, but when you view the highway data, you can see that it does not form a full circle. Why not?

Version: 20-Apr-2022 5PM

## Change Notes

1. TBA