Assignment 2

COSC 6320

~~This assignment is being developed. Students should start working on the questions. Check back often for additional questions. Once the last question is posted, students will have two days to turn in the whole assignment.~~

This file was modified on 11/22.

[11/23] This is the final version of the assignment set. Due 11/29.

1. [B-trees] This assignment wants to verify if the Fringe Analysis accurately estimates 2-3 tree storage utilization correctly.

   (a) You are asked to build 2-3 trees using the same assumptions described in the class. You should include a function to initialize an empty 2-3 tree, an insertion function, a traversal function to collect statistics. The traversal algorithm will count the number of 2-nodes and the number of 3-nodes in the tree. Also count the number of type-i subtrees (i = 1, 2, …, 7). With that, you can compute the storage utilization of the tree. You will be given a file of 1000 randomly generated integer numbers (which may not be unique). Collect the statistics after every 100 insertions. The TA will test your program with other data sets.

   (b) Compare the theoretical result from the analysis and the experiments. For this part, you will write some design and conclusion of the experiment. Are the type-*i* subtrees as predicted? Is the storage utilization within the range? Is it converging? Whatever you can get from the investigation. The program is just part of the assignment. You must write a discussion (at least one page) on the design of the experiment. Also, tell us what is learned from this experiment.

   The test data file is available now. It contains 1000 randomly generated comma-separated integer numbers between 0 and 10,000. There may be duplicated numbers among them. You should keep all numbers in the 2-3 tree. And make sure you will be able to find them later. The TA will test the code with other sets of data.

2. [Directed Graph] Given a directed graph with positive costs associated with the edges, design an algorithm to find the shortest path from v1 to v2. You can use the all-pair or single-source shortest path to answer this question, but the cost is maybe unnecessarily high. Find the most efficient algorithm to do so. Write a program for this algorithm. You can design how the graph is implemented and what data structure to use to make the algorithm more efficient. Show the complexity of your algorithm and justify it. You may also prove that the algorithm is correct.

   The program of the most efficient algorithm is just part of the assignment. You must write a discussion on the design of the solutions, including all the algorithms that solve this problem. You don't have to code all of them, just the most efficient one. Again, I do expect at least a one-page discussion.

   A test data file has been published. The first line of the text file contains the size (n) of the graph. The following n lines contain the adjacency matrix of the directed graph. They are arranged one row (row

0 to n-1) per line separated by commas. The final line includes two vertices v1 and v2. The TA will test the code with other sets of data.

3. What would you recommend if you were asked to design a data structure for storing vaccination records (for vaccine passport use, perhaps)? Please consider all the data structures we discussed in class and consider the possibility of using more than one data structure to speed up the operation. Write the design, the analysis (that support your design), and the justifications, including data storage and efficiency. This assignment asks for your design, and no code is needed. There is no one correct answer.

The following assignments are for your final exam preparation. Do not submit the solutions.

[Quick Sort] Implement the sorting algorithm. What is the worst-case complexity?

[Graph Traversal] Implement the Depth First Search (DFS) algorithm.

[Algorithm Design Techniques] Divide and Conquer, Dynamic Programming, Greedy, Backtracking, etc.