University of Nevada, Las Vegas Computer Science 477/677 Fall 2020

Assignment 7: Due Monday November 9, 2020

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. Your answers must be written in a pdf file and emailed to the graudate assistant, Tandreana Chua `chuat4@unlv.nevada.edu` , by midnight November 5. Your file must not exceed 10 megabytes, and must print out to at most 8 pages.

Unless otherwise specified, n refers to the number of vertices and m refers to the number of edges of a graph or directed graph.

1. Give an algorithm for converting the array of outneighbor lists of a weighted directed graph into the array of inneighbor lists of the same graph in $O(n + m)$ time.

2. 8 items need to be inserted into a cuckoo hash table of size 8. Insert the items in the order listed below. After each item, there are two hash values.

Ann	0	5
Bob	5	2
Cal	2	4
Dan	4	0
Eve	3	6
Fay	1	7
Gus	4	3
Hal	5	4

3. Give code or pseudocode for the longest monotone increasing subsequence problem. Given a sequence of n real numbers A(1) ... A(n), determine a subsequence (not necessarily contiguous) of maximum length in which the values in the subsequence form a strictly increasing sequence. (I have a Youtube video for this problem.)

4. Give an $O(n + m)$-time algorithm for finding a topological order, if there is one, for a directed graph given as an array of inneighbor lists. If the graph does not have a topological order, your algorithm should report that fact and find a cycle.

5. Suppose that you wish to store a $n \times n \times n$ 3-dimensional array A of integers. You could declare int `A[n][n][n]`, using n^3 space. However, in your application, you only need values of `A[i][j][k]` for i $\geq$ j $\geq$ k. Explain how you would construct a data structure for A which takes $n^3/6 + O(n^2)$ space, and which allows the operators **fetch** and **store** to be executed in $O(1)$ time each.

6. Starting with an empty AVL tree, insert the items A, B, F, E, D, C in that order. Show each step, including the rotations.

7. A treap is a binary tree where the nodes are ordered in two ways. Each item has a *value* (say ”Sam”) and a randomly chosen *priority*. The nodes are ordered so that the values are alphabetic, as in a normal binary search tree, and the priorities are in heap order, just as in the binary tree implementation of a

heap. Create a treap with the following items, inserted in the order given, where the priority of each item is given in the second column. For ease of grading, make it a min-heap. Show the steps, including the rotations.

Ann	12
Bob	10
Eve	17
Dan	1
Cal	19
Ted	4
Sue	5

8. Find a page on the internet which explains the Collatz conjecture. For any positive number n, let f(n) = n/2 if n is even, 3n+1 if n is odd. The conjeccture is that, if you start with any positive integer, you will reach 1 after applying f finitely many times. That number of times is called the *total stopping time* of n.

   For example, f(10) = 5, since the sequence is 10, 5, 16, 8, 4, 2, 1. Here is a recursive function which computes total stopping time:

```
int TST(int n)
  {
   assert(n > 0);
   if (n == 1) return 0;
   else if (n%2) return TST(3*n+1);
   else return TST(n/2);
  }
```

   Describe a program to find the total stopping times of all positive integers from 1 to 100, using memoization. Why is memoization better than either recursion or dynamic programming for this problem? Hint: try computing TNT(27).