AUB-CMPS 256: *Advanced Algorithms and Data Structures* December 16th, 2013

Final exam – Fall 2013-14

Instructor: Fatima Abu Salem **Name:**

Duration: 90 minutes

This Exam is Closed-Notes. It has four questions. Make sure you read all of them before you begin writing your solutions.

Question 1 (18%)

On Linear Time Sorting:

(a) (9%) Think about a real life application that requires sorting, but which you can achieve in linear time using Counting Sort. Make sure to describe the relevant parameters characterising your input size, and thus why, Counting Sort, is guaranteed to run using a linear function of the number of elements comprising the input.

(b) (9%) Give a numerical example of a list of integers to be sorted where Counting Sort would be extremely inefficient. Explain the source of inefficiency.

Answer Sheet for Question 1

Answer Sheet for Question 1

Question 2 (25%)

On Priority Queues Using a Max-heap

Suppose you're given a list of integers that is known to contain duplicates. Suppose you also build a Max-heap using this list of integers.

(a) (8%) Let max denote the value of the largest integer appearing in the list. Where would the duplicates of max reside in the heap?

(b) (8%) Suppose now that there are duplicates of another integer that is not equal to max. Where would those duplicates reside in the heap?

(c) (8%) Would you recommend using a Max-heap priority queue for a generic (random) search query? Justify why or why not.

Answer Sheet for Question 2

Answer Sheet for Question 2

Question 3 (18%)

On Binary Search Trees

(a) (9%) Given a random binary tree instantiated using nodes (i.e. relying on pointers rather than index arithmetic), develop a recursive algorithm to return whether this given binary tree is a BST.

(b) (9%) Describe in detail the run-time of this algorithm.

Answer Sheet for Question 3

Answer Sheet for Question 3

Question 4 (40%)

On Graph Algorithms:

(a) (8%) Given a directed, unweighted graph $G = (V, E)$, develop an alogrithm which returns whether or not G has a cycle. State the run-time of this algorithms.

(b) (8%) Consider the arithmetic expression E given by $5 + 2 * 4 + (7 * 4)/2$. Recall the rules governing order of precedence in arithmetic operations. For example, first in the chain of execution should be $(7*4)$, followed by its division by 2, and so on and so forth. Construct a graph which represents all the dependencies in the arithmetic operations required by E.

(c) (8%) Describe a linear-time algorithm for evaluating E and justify the run-time.

(d) (8%) Consider a variation of the Minimum Spanning Tree problem where we're given an undirected, weighted graph $G = (V, E)$, all weights being positive, and we seek to minimise the product of edge weights, taken over some spanning tree. Describe a simple modification to the standard MST algorithm we have seen in class that will return a minimum product spanning tree.

(e) (8%) Consider a weighted, directed graph $G = (V, E)$ with no negative weight cycles. Recall the greedy choice property the Dijkstra's algorithm makes in all of its iterations. Suppose that an adversary is able to return a shortest paths tree T without making a greedy choice at some point in time. Show that someone using Dijkstra's algorithm can still produce a tree T' that contains the greedy choice.

Answer Sheet for Question 4

Answer Sheet for Question 4

Answer Sheet for Question 4

Answer Sheet for Question 4