CS 5633 Analysis of Algorithms – Fall 16

Exam 3

NAME:

- This exam is closed-book and closed-notes, and electronic devices such as calculators or computers are not allowed. You are allowed to use a cheat sheet (half a single-sided letter paper).
- Please try to write legibly – if I cannot read it you may not get credit.
- **Do not waste time** – if you cannot solve a question immediately, skip it and return to it later.

1) Greedy Algorithms		30
2) Amortized Analysis		30
3) Graph Algorithms		20
4) Minimum Spanning Trees/Shortest Paths		20
		100

1 Greedy Algorithms (30 Points)

Suppose we are managing a zoo with animals roaming around a large open field that can be partitioned into a West portion and an East portion. The zookeepers want to be able to determine when an animal wanders from one region to the other, crossing a long imaginary line cutting through the field. To detect this, they want to activate several sensors that can detect when some interval of this region has been crossed. Sensors cannot be moved, we can only decide if we want to activate them or not. We want to minimize the number of sensors that together cover the entire line. Give a greedy algorithm to solve this problem, and prove why your answer is correct.

2 Amortized Analysis(30 Points)

Recall the dynamic array data structure that was asked in class. If our data structure currently has size k, then we double the size to $2k$ once we receive operation $k+1$ and copy all of the elements from the old data structure to the new one. Suppose that instead of doubling to size $2k$, we add 100 to obtain an array of size $k+100$. This question will consider the amortized cost of an operation in this data structure.

1. Perform an aggregate analysis to obtain a bound on the sum of the cost of n operations.

2. Use the accounting method to determine the amortized cost of a single operation.

3 Graph Algorithms (20 Points)

A directed graph is *weakly connected* if for every pair of vertices u, v we have that either there is a path from u to v or there is a path from v to u. Give an algorithm to determine if a given directed graph G is weakly connected. Make your algorithm as efficient as possible.

4 Minimum Spanning Tree/Shortest Paths (20 Points)

In class we showed how to compute a minimum spanning tree and a shortest path in a graph. Suppose we want to do the opposite for both of them. That is, suppose we want to compute a Maximum Spanning Tree (spanning tree that maximizes sum of edge weights) and a Longest Path (path in the graph that maximizes the sum of edge weights where we visit each vertex once). Can we easily modify the algorithms that we have seen in class to handle these cases? If so how can this be done? If not then what is the difficulty?