EE360C: Algorithms
The University of Texas at Austin — Exam #2
Dr. Pedro Santacruz & Dr. Evdokia Nikolova — November 3, 2014

Name: **EID:**

Exam #2

Instructions. No calculators, laptops, or other devices are allowed. This exam is **closed book**, but you are allowed to use a **one-page** cheat sheet. You must submit your cheat sheet with the exam. Write your answers on the test pages. If you need scratch paper, use the back of the test pages, but indicate where your answers are. Write down your process for solving questions and intermediate answers that **may** earn you partial credit.

If you are unsure of the meaning of a specific test question, write down your assumptions and proceed to answer the question on that basis. **Questions about the meaning of an exam question will not be answered during the test.**

You have **75 minutes** to complete the exam. The maximum possible score is 100.

Some useful information:

Logarithms and Factorial:

$$\log(n!) = \Theta(n \log n)$$

Arithmetic Series:

$$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1)$$

Sum of Squares:

$$\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

Sum of Cubes:

$$\sum_{k=0}^{n} k^3 = \frac{n^2(n+1)^2}{4}$$

Geometric Series:

$$\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$$

Infinite Geometric Series:

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$$

Problem 1: Minimum Spanning Trees [20 points]
Let $G = (V, E)$ be an undirected graph. Each edge $e \in E$ has a *distinct* weight w_e, that is, for every two edges $e_1, e_2 \in E$, if $e_1 \neq e_2$, then $w_{e_1} \neq w_{e_2}$.

Prove or disprove the following: There is a unique minimum spanning tree in G.

Problem 2: Greedy Algorithm [20 points]
Consider the problem of matching a set of available skis to a set of skiers. The input consists of n skiers with heights $p_1, \ldots p_n$ and n sets of skis with lengths $s_1, \ldots s_n$. The problem is to assign each skier a set of skis to minimize the average difference between the height of a skier and the length of his or her assigned set of skis. That is, if the i^{th} skier is given the $\alpha(i)^{th}$ pair of skis, then you want to minimize:

$$\frac{1}{n}\sum_{i=1}^{n}|p_i - s_{\alpha(i)}|$$

Consider the following greedy algorithm:

> Find the difference between the height of each skier and the length of each set of skis. Choose the smallest *difference* first, and assign those skis to that skier. Continue to make the next greediest choice until all skiers have been assigned skis

Prove or disprove that this greedy choice is optimal.

Problem 3: Shortest Paths [20 points]
You are planning a trip from Austin to New York City. While doing your research to get the cheapest airfare possible you realize that flying directly to New York might not be the best way to go since your main concern is cost. Below are the cost of flights between different cities.

a) Show step by step execution (do not write the original algorithm) of Dijkstra's algorithm on the given graph to find the cheapest flight from Austin to New York.

b) List the path (a separate path for each city, your answer should list nine paths in total) of cheapest flights from Austin to each of the remaining cities.

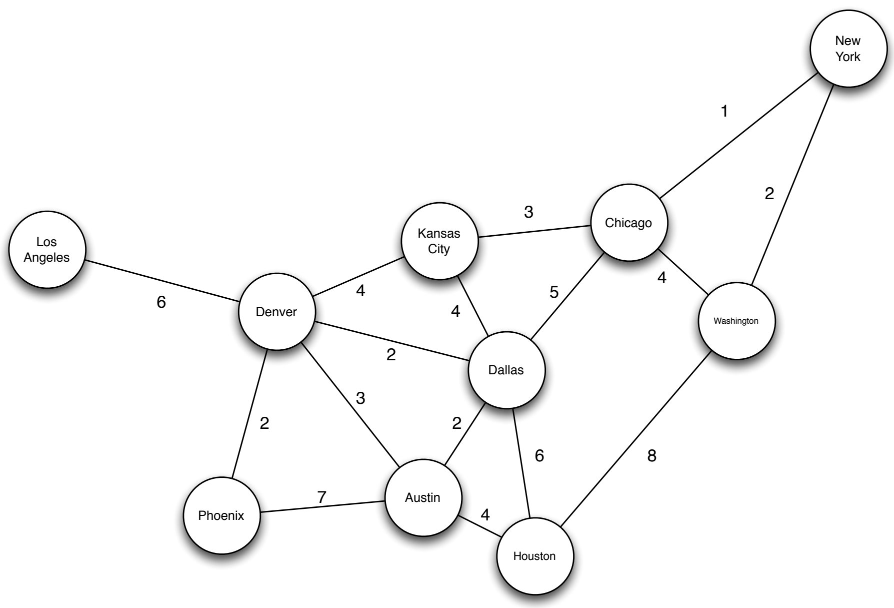

Figure 1: Cost of air travel between US cities

Problem 4: Divide and Conquer [20 points]

At a particular store, they've been doing a terrible job of managing their staff, specifically with respect to having the right number of staff on hand at the busiest times. After just a few days on the job, you notice that the crowds follow a very specific pattern: the number of customers strictly increases up to a point and then strictly decreases. The shop's registers give you a print out of the average number of customers in every ten minute interval from open to close.

Given this array of the customer counts over a given day, write an efficient algorithm to find the specific 10 minute interval that is the "peak" traffic for your coffee shop. Give as efficient an algorithm as possible. What is the running time of your algorithm?

Problem 5: Directed Acyclic Graphs [20 points]
Consider an unweighted directed acyclic graph (DAG) $G = (V, E)$ with n vertices and m edges. Give an $O(m + n)$ algorithm to find the longest path in G.

1)	2)	3)	4)	5)	Total

Scratch Page