

## CS 330 – Fall 2009, Assignment 3

Problems due at the beginning of class on Thursday, October 15  
Late assignments will only be accepted up until 2PM Friday, October 16

**Question 1** (10 pts). Chapter 4, Exercise 19, on p. 198.

**Question 2** (10 pts). Consider the single-source shortest paths problem where all edge costs are integers between 1 and  $C$ . Describe an implementation of Dijkstra's algorithm that works in time  $O(m + Cn)$ . Hint: Look at Proposition 4.15 on p. 142.

**Question 3** (10 pts). Chapter 5, Exercise 5, on p. 248.

**Question 4** (10 pts). Chapter 5, Exercise 6, on p. 248.

**Question 5** (20 pts). In this programming question, we will be considering MSTs on complete, undirected graphs. A graph with  $n$  vertices is complete if all possible  $\binom{n}{2}$  edges are present in the graph. Consider the following two types of graphs:

- Complete graphs on  $n$  vertices, where the weight of each edge is a real number chosen uniformly at random from  $[0, 1]$ .
- Complete graphs on  $n$  vertices, where the vertices are points chosen uniformly at random inside the unit square. (That is, the points are  $(x, y)$ , with  $x$  and  $y$  each a real number chosen uniformly at random from  $[0, 1]$ .) The weight of an edge is the Euclidean distance between its endpoints.

Your goal is to determine how the average weight of the minimum spanning tree grows as a function of  $n$  for each of these families of graphs. You will need to implement an MST algorithm and procedures that generate the appropriate random graphs. Run your program on at least five random graph instances for

$$n = 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192$$

and report the average for each value of  $n$ . Interpret your results by giving a sample function  $f(n)$  that describes your plot. For example, your answer might be  $f(n) = 2 \log n$ ,  $f(n) = 1.5\sqrt{n}$ , etc. Also, provide some intuition for why the growth rate of your functions  $f(n)$  are reasonable. Do this for both types of graphs separately.

You may work in teams of up to three people for this question. Teams of two are probably ideal. Send me e-mail if you'd like me to set you up with a random partner. Each team must submit one hard copy of their code along with one joint writeup for this question.