Homework 3

CS 4104 (Spring 2016)

Assigned on Tuesday, February 16, 2014.
Submit a PDF file containing your solutions on Canvas by the beginning of class on Thursday, February 25, 2014.

Instructions:

- You can pair up with another student to solve the homework. You are allowed to discuss possible algorithms and bounce ideas with your team-mate. **Do not discuss proofs of correctness or running time in detail with your team-mate.** Please form teams yourselves. Of course, you can ask me for help if you cannot find a team-mate. You may choose to work alone. *Each of you must write down your solution individually, and write down the name of the other member in your team. If you do not have a team-mate, please say so.*
- Apart from your team-mate, you are not allowed to consult any sources other than your textbook, the slides on the course web page, your own class notes, the TAs, and the instructor. In particular, do not use a search engine.
- Do not forget to typeset your solutions. *Every mathematical expression must be typeset as a mathematical expression, e.g., the square of n must appear as n^2 and not as "n^2".* Students can use the LaTeX version of the homework problems to start entering their solutions.
- Describe your algorithms as clearly as possible. The style used in the book is fine, as long as your description is not ambiguous. Explain your algorithm in words. A step-wise description is fine. *However, if you submit detailed pseudo-code without an explanation, we will not grade your solutions.*
- Do not make any assumptions not stated in the problem. If you do make any assumptions, state them clearly, and explain why the assumption does not decrease the generality of your solution.
- Do not describe your algorithms only for a specific example you may have worked out.
- You must also provide a clear proof that your solution is correct (or a counter-example, where applicable). Type out all the statements you need to complete your proof. *You must convince us that you can write out the complete proof. You will lose points if you work out some details of the proof in your head but do not type them out in your solution.*
- Describe an analysis of your algorithm and state and prove the running time. You will only get partial credit if your analysis is not tight, i.e., if the bound you prove for your algorithm is not the best upper bound possible.

Problem 1 (20 points) Solve exercise 2 in Chapter 4 (page 189) of your textbook. *Note:* We will discuss minimum spanning trees in the next class or two, but you can read the definition on page 142 of your textbook.

For each of the following statements, decide whether it is true or false. If it is true, give a short explanation. If it is false, give a counterexample. In both problems, we are given a graph G, with edge costs that are all positive and distinct.

(a) Let T be a minimum spanning tree of G. Suppose we replace each edge cost c_e by its square c_e^2, thereby creating a new graph with the same edges but different costs. True of False? T must still be a minimum spanning tree for this new graph. *Note:* The total cost of the edges in a minimum spanning tree will change, of course. We are asking if the set of edges of T must also change when we square edge costs.

(b) Let P be a minimum-cost s-t path. Suppose we replace each edge cost c_e by its square c_e^2, thereby creating a new graph with the same edges but different costs. True of False? P must still be a minimum-cost s-t path in this new graph.

Problem 2 (35 points) Solve exercise 5 in Chapter 4 (pages 190–191) of your textbook. Let's consider a long, quiet country road with house scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let's suppose that despite the bucolic setting, the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

Given an efficient algorithm that achieves this goal, using as few base stations as possible.

Just in case the problem statement is not completely clear, you can assume that the road is the x-axis, that each house lies directly on the road, and that the position of each house can be specified by its x-coordinate.

Problem 3 (45 points) Solve exercise 13 in Chapter 4 (pages 194–195) of your textbook. A small business—say, a photocopying service with a single large machine—faces the following scheduling problem. Each morning, they get a set of jobs from customers. They want to do the jobs on their single machine in an order that keeps the customers happiest. Customer i's job will take t_i time to complete. Given a schedule, i.e., an ordering of jobs, let C_i denote the finishing time of job i. For example, if job j is the first to be done, we would have $C_j = t_j$, and if job j is done right after job i, then $C_j = C_i + t_j$. Each customer i also has a given weight w_i that represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of i's job. So the company decides to order the jobs to minimize the weighted sum of completion times, $\sum_{i=1}^{n} w_i C_i$.

Design an efficient algorithm to solve this problem, i.e., you are given a set of n jobs with a processing time t_i and weight w_i for each job. You want to order the jobs so as to minimize the weighted sum of completion times, $\sum_{i=1}^{n} w_i C_i$.

Hint: Try to use one of the techniques we have seen for proving the correctness of greedy algorithms. Working "backwards" from what you need to prove might help you to discover the algorithm.