

## Assignment 3 solutions

2a) True, squaring the weights of the edges in a weighted graph will not change the minimum spanning tree. Assume the opposite to obtain a contradiction. If the minimum spanning tree changes then at least one edge from the old graph  $G$  in the old minimum spanning tree  $T$  must be replaced by a new edge in tree  $T'$  from the graph  $G'$  with squared edge weights. The new edge from  $G'$  must have a lower weight than the edge from  $G$ . This implies that there exists some weights  $C_1$  and  $C_2$  such that  $C_1 < C_2$  and  $C_1^2 \geq C_2^2$ . This is a contradiction.

2b) False, squaring the weights of the edges in a directed graph may change the shortest path between two points. Consider the following graphs:



It is clear that in the version with the edges squared the shortest path from  $a$  to  $c$  has changed. Some people thought that the proof from (2a) could also apply to this problem. However, while the sorted order of edges is all that matters in finding the minimum spanning tree and this is not affected by squaring, when finding a shortest path the ratio between the edges is important and this is affected by the squaring function.

3) I think this problem was stated in a slightly confusing way. To minimize the number of trucks used, one should ship the packages from smallest to largest. Consider three packages with sizes 5%, 98%, and 5% of a trucks capacity; shipping them in the order they arrive requires three trucks, clearly a waste. However, we were not asked to judge the relative value of customer satisfaction. With the constraint that packages must be shipped in the order they are received, the optimal algorithm is a greedy one: pack each truck as fully as possible. To see that delaying the delivery of some packages in an effort to more fully pack later trucks is not more efficient, we only need to observe that this strategy will merely increase the number of boxes to be shipped. Since this algorithm will never free room on a later truck so that it can carry additional boxes, there is no way for it to outperform the greedy approach.

5) This problem also has a simple greedy solution which nearly everyone found:

- Place a cell tower 4 miles east of the westernmost house
- Continue to the east placing a cell towers 4 miles east of the first house more than 4 miles east of the last cell tower.  
Stop when there are no remaining houses more than 4 miles east of the last cell tower or when the eastern endpoint of the road is less than 4 miles from the westernmost house more than 4 miles east of the last cell tower.
- If the easternmost house is more than 4 miles from a cell tower, place a tower at that house.

7) The solution is to queue the problems in descending order of  $f_i$ . That is, from longest to shortest running time on the PCs. Since every job must be run consecutively on the supercomputer, the preprocessing job times will be summed regardless of our ordering. The worst possible completion time would be if we started the longest  $f_i$  after the last job finishes in the supercomputer. That is,

$$CompletionTime \leq \sum p_i + [f_i]$$

Thus, if we begin the longest PC job first we are sure to minimize the amount of time the PCs are operating after the preprocessing is done.