CMPT 705 — Design and Analysis of Algorithms

Exercises on Graphs and Greedy Algorithms. Due: Tuesday, September 27th (at the beginning of the class)

Reminder: the work you submit must be your own. Any collaboration and consulting outside resourses must be explicitely mentioned on your submission.

1. Consider the problem of making change for n cents using the fewest number of coins. Assume each coin's value is an integer.

   (a) Describe a greedy algorithm to make change consisting of quaters, dimes, nickels, and pennies. Prove that your algorithm is optimal.

   (b) Give a set of coins for which the greedy algorithm does not yield an optimal solution. Your set should include a penny so that there is a solution for every value of n.

2. In social networks analysis it is sometimes useful to find not only the distance between two vertices in a graph or a shortest path between them, but the number of such paths. It turns out this problem can be solved efficiently. More precisely, given an undirected graph (no lengths) $G = (V, E)$ with $|V| = n$ and $|E| = m$, and two vertices $v, w \in V$, suggest an algorithm that outputs the number of shortest $v - w$-paths in G. (The algorithm should not list all the paths, just the number will do.) The running time of your algorithm should be $O(m + n)$.

3. You are consulting for a trucking company that does a large amount of business shipping packages between New York and Boston. The volume is high enough that they have to send a number of trucks each day between the two locations. Trucks have a fixed limit W on the maximum amount of weight they are allowed to carry. Boxes arrive at the New York station one by one, and each package i has a weight w_i. The trucking station is quite small, so at most one truck can be at the station at any time. Company policy requires that boxes are shipped in the order they arrive; otherwise, a customer might get upset upon seeing a box that arrived after his make it to Boston faster. At the moment, the company is using a simple greedy algorithm for packing: they pack boxes in the order they arrive, and whenever the next box does not fit, they send the truck on its way.

   But they wonder if they might be using too many trucks, and they want your opinion on whether the situation can be improved. Here is how they are thinking. Maybe one could decrease the number of trucks needed by sometimes sending off a truck that was less full, and in this way allow the next few trucks to be better packed.

   What is your opinion? If you think the algorithm is optimal, prove it. If you think it is not, give a set of boxes with specified weights, that are packed in a non-optimal way.

4. Let us say that a graph $G = (V, E)$ is a *near-tree* if it is connected and has at most $n + 8$ edges, where $n = |V|$. Give an algorithm with running time $O(n)$ that takes a near-tree G with weights on its edges, and returns a minimum spanning tree of G. You may assume that all the edge weights are distinct.