CSCI211: Problem Set 5

Due Friday, March 2
Possible Points: 27

1. **(6 pts) (4.3) 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 one 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.

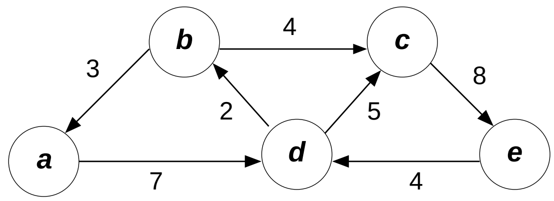
   Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed. Your proof should follow the type of analysis we used for the Interval Scheduling Problem: it should establish the optimality of this greedy packing algorithm by identifying a measure under which it "stays ahead" of all other solutions.

2. **(6 pts) You and your roommates, all at least 21 years of age, have gotten hooked brewing your own beer from kits. You've made quite a few batches, each from a different mix, so you now have an impressively varied range of flavors: summer cherry wheat, chocolate hazelnut lager, pepperoni pale ale, etc. You'd like to get through all the batches before your court-mandated AA meetings are set to begin. You can probably polish off a batch a day, but some batches are clearly better than others, and you can't decide in which order to drink them. Further complicating matters, each beer is aging, causing its quality to change over time. Sometimes, like a good wine, this change is for the better. Other times, as in the case of the pepperoni pale ale, each day you wait makes it that much more likely to kill you. Of course, you're not willing to throw anything out, so you've got a little optimizin' to do.**

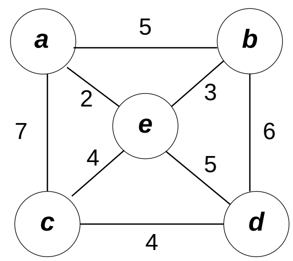
   To make things concrete, you have n batches of beer, which you'd like to drink over the next n days, one batch per day. Each batch i has an initial quality q_i and a rate of change r_i, indicating how much that quality changes each day. So, if you drink beer i after d days, the value you get will be $q_i + d \cdot r_i$. Your goal is to maximize the sum of values over all n beers. Keep in mind that r_i may be negative for some i, but you still intend to consume all the beer in n days, even if the value for some beers is negative.

   Give an efficient algorithm to solve this problem, and prove it is optimal.

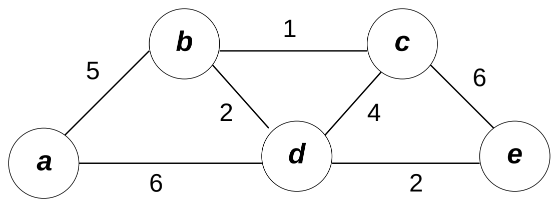
3. **(3 pts) [Levitin 9.3.2] Solve the following for the single-source shortest-paths problem with vertex a as the source. (Show your work.)**



4. **(3 pts) [Levitin 9.1.7] Apply Prim's algorithm to the following graph. (Show your work.)**



5. **(3 pts) [Levitin 9.2.1] Apply Kruskal's algorithm to find a minimum spanning tree of the following graph. (Show your work.)**



6. **(6 pts) [K&T 4.2] For each of the following two statements, decide whether it is true or false. If it is true, give a short explanation. If it is false, give a counter example.**

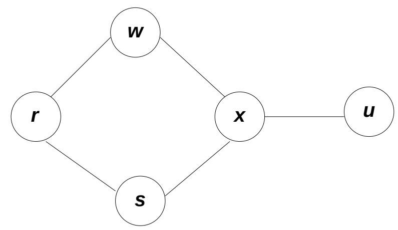
   (a) **Suppose we are given an instance of the Minimum Spanning Tree Problem on a graph G, with edge costs that are all positive—actually, ≥ 1—and distinct. Let T be a minimum spanning tree for this instance. Now suppose we replace each edge cost c_e by its square, c_e^2, thereby creating a new instance of the problem with the same graph but different costs.**

   True or false? T must still be a minimum spanning tree for this new instance.

   (b) **Suppose we are given an instance of the Shortest $s - t$ Path Problem on a directed graph G. We assume that all edge costs are positive—actually, ≥ 1—and distinct. Let P be a minimum-cost $s - t$ path for this instance. Now suppose we replace each edge cost c_e by its square, c_e^2, thereby creating a new instance of the problem with the same graph but different costs.**

   True or false? P must still be a minimum-cost $s - t$ path for this new instance.

7. **Bonus, 5 pts. A valid k-coloring of a graph is an assignment of colors to nodes such that adjacent nodes are assigned different colors. For example, there is a 2-coloring of the graph below that colors vertices x and r red, and colors the remaining vertices s, w and u blue.**



Suppose you have a graph $G = (V, E)$ and you'd like to color it using as few colors as possible. Consider the following greedy algorithm that attempts to find a minimum coloring of G (using numbers as colors):

Order the vertices from highest to lowest degree, and initially have all nodes unlabeled. For each node v in our ordering, assign v the smallest positive integer that does not conflict with any neighbor of v.

(a) **Run the algorithm on the graph above; what coloring does it find?**

(b) **Show that this algorithm does not always find a minimum coloring.**

(c) **Prove that this greedy algorithm always finds a coloring using at most $\Delta(G) + 1$ colors, where $\Delta(G)$ is the maximum degree of any vertex in G.**