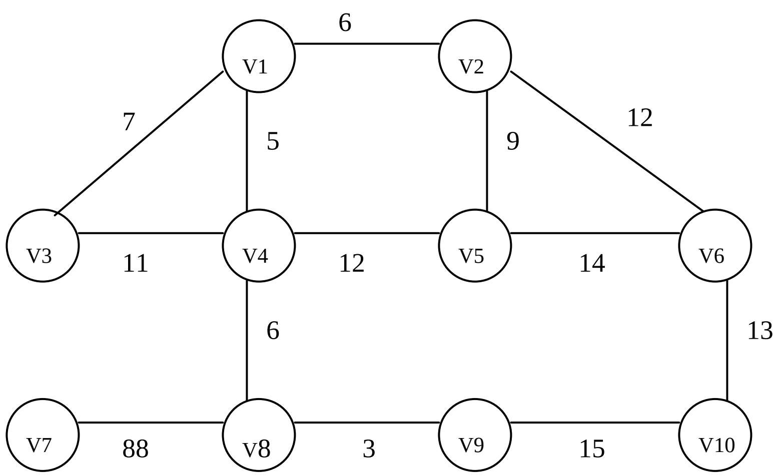
CS 331 Design and Analysis of Algorithms
Homework #3
(Due: 3/4/2014)

Note: Turn in your homework with a cover sheet including your name and last 4 digits of your student ID # so that your grade can be correctly recorded. Write on only one side of each page.

(Total: 100 points)

1. (20 points)
Use Prim's Algorithm to find
(a) a minimum spanning tree and
(b) the cost of the minimum spanning tree for the following graph.
Show all steps. (Assume that the algorithm starts from v1)



2. (20 points)
For the same graph given in question 1, use Dijkstra's Algorithm to find
(a) lengths of shortest paths from the source v1 to all the remaining vertices of the graph.
(b) shortest paths from the source v1 to all the remaining vertices of the graph.
Show all steps.

3. (20 pints)

[0/1 Knapsack] Consider the knapsack problem discussed in class. We now add the requirement that x_i=1 or x_i=0, $\forall$ *i*. That is, an object is either completely included or not included into the knapsack. We call this new version of knapsack problem as *0/1 Knapsack problem,* and we still want to maximize the profit subject to the total weight $\leq M$.

(a) One greedy strategy is: consider the objects in order of nonincreasing density p_i/w_i; add the object into the knapsack if it fits. Show that this strategy doesn't necessarily yield optimal solutions.

(b) One greedy strategy is: consider the objects in order of nondecreasing density p_i/w_i; add the object into the knapsack if it fits. Show that this strategy doesn't necessarily yield optimal solutions.

4. (20 points)

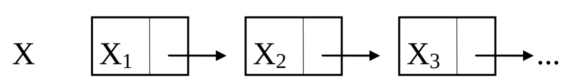
Let A_n = { a_1, a_2, ..., a_n } be a finite set of distinct coin types (e.g., a_1= 25 cents, a_2= 10 cents, a_3= 5 cents etc.). We assume each a_i is an integer and that $a_1 > a_2 > ... > a_n$. Each type is available in unlimited quantity. The coin changing problem is to make up an exact amount C using a minimum total number of coins. C is an integer > 0.

(a) When $a_n = 1$ a greedy solution to the problem will make change by using the coin types in the order a_1, a_2, ... , a_n. When coin type a_i is being considered, as many coins of this type as possible will be given. Write an algorithm based on this strategy.

(b) Give a counterexample to show that the algorithm in (a) doesn't necessarily generate solutions that use the minimum total number of coins.

(c) Show that if A_n = { k^{n-1}, k^{n-2}, ... , k^0 } for some k > 1 then the greedy method in (a) always yields solutions with a minimum number of coins.

5. (20 points) Let X be a set of n elements to be maintained as a linked list:



Assume that the cost of examining a particular element X_i is C_i. Note that to examine X_i, one needs to scan through all elements in front of X_i. Let P_i be the probability of searching for element X_i, so the total cost for all searches is

$$\sum_{j=1}^{n} \left[P_j \bullet \sum_{i=1}^{j} C_i \right]$$

(a) Show that storing elements in nonincreasing order of P_i/C_i does not necessarily minimize the total cost.

(b) Show that storing elements in nondecreasing order of P_i does not necessarily minimize the total cost.