Assignment #4, CS/531

Due Date: Mon. Nov. 7, 2011

UNSUPPORTED SOLUTIONS RECEIVE NO CREDIT.
Total points: 51

1 (7 pts). Maximum Contiguous Subsequence Sum Problem revisited.

Let $A[1..n]$ be an array of numbers. The elements in A can be either positive or negative. We want to find the indices k, l so that the sum $\sum_{i=k}^{l} A[i]$ is maximum among all possible choices of k, l. For example if $A = \{-3, 12, -6, 10, -5, 2\}$, the answer is $k = 2, l = 4$, since $A[2] + A[3] + A[4] = 12 + (-6) + 10 = 16$ is the maximum sum of all possible choices.

This is exactly the problem 3 in HW2. However, this time you must describe an $O(n)$ time algorithm for solving this problem.

2. (1 pt) Consider the graph $G = (V, E)$ shown in Figure 1. The integers near an edge is its weight. Using Kruskal's algorithm, compute the minimum spanning tree T_1 of G. List the edges of T_1 **in the order they are added into** T_1.

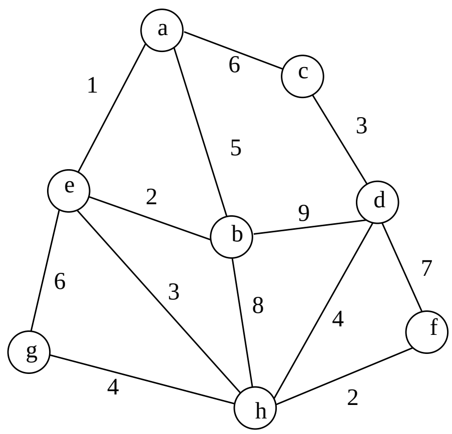

Figure 1: Kruskal's Algorithm.

3. (6 pts) For a given undirected, edge weighted graph $G = (V, E)$, there might be more than one MST of G. This can happen if G has two edges e' and e'' such that $w(e') = w(e'')$. In this case, when we run Kruskal algorithm on G, the edges e' and e'' may be processed in different order, and this might results in two different MSTs of G.

Suppose that all edge weights of G are different. (Namely for any two edges e' and e'', we have $w(e') \neq w(e'')$.) Under this condition, show that G has a unique MST.

4. (7 pts) Consider a long, quite country road with houses scattered very sparsely along it. You may picture the road as a long straight line segment, with the western endpoint (mile stone 0), and the eastern endpoint (mile stone L). Each house is identified by its distance to the western endpoint. A cell phone company wants to set up cell phone services along the road. The company can place a base station at any house. (The monthly charge will be waived if a base station is located in a house, so the house owners are eager to accommodate base stations). The power of base stations are limited that can only cover a distance of 5 miles. The goal for the company is to select a minimum number of base stations so that every house on the road

is within 5 miles of a base station. (The bases stations are connected by other means, say by Satellite. So the distance between them can be more than 5 miles).

A formal description of the problems: The input is a set X of n points: $X = \{x_1 < x_2 < \ldots < x_n\}$, where each x_i $(1 \leq i \leq n)$ represents a house. We need to select a subset $Y \subseteq X$ such that: (1) for every point $x_i \in X$, there is a point $x_j \in Y$ with $|x_i - x_j| \leq 5$, and (2) the size of Y is minimum, subject to condition (1).

Describe a greedy algorithm for solving this problem. Argue why your algorithm produces an optimal solution and analyze it's runtime.

5. $(2 + 5 = 7$ pts) We want to design a set of weights $\{w_1, w_2, \ldots, w_n\}$ for a balance scale. Each w_i is an integer (say w_i grams). The choice of weights must ensure that, for every integer load $k \in \{1, 2, \ldots, W\}$, we can measure the load by using a combination of the weights. The goal is to maximize the range $[1..W]$ (for the given fixed n). There are two possible situations:

(a) The weights can only be put on the free cup of the scale. (In this setting, if we have $n = 4$, $w_1 = 1, w_2 = 2, w_3 = 2, w_4 = 5$, then we can measure every integer load in the range $[1..10]$.)

(b) The weights can be put on both cups of the scale. (For example, if we put a weight $w_1 = 1$ on one cup, and $w_2 = 3$ on the other cup, we can measure a load of 2 grams).

In either setting, describe the values of $w_1, \ldots, w_n$, and the maximum range W. Argue why every integer load $k \in \{1, \ldots, W\}$ can be measured. (This is not really an algorithm design problem. Rather, it is a *greedy way of thinking*).

6. (4 pts) Page 593, Problem 22.1-5.

7. (2 pts) Show the $d(*)$ and $\pi(*)$ values that result from running BFS (bread-first search) algorithm on the graph shown in Figure 2, using vertex a as the starting vertex. (Assuming the adjacency lists are arranged in alphabetical order).

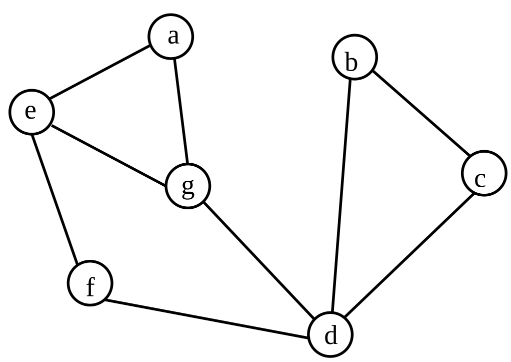

Figure 2: BFS algorithm.

8. (2 pts) Show how the DFS (depth-first search) algorithm works on the directed graph shown in Figure 3. Assume that the for loop of DFS procedure considers the vertices in alphabetical order and that the adjacency-lists are ordered by alphabetical order. Show the discovery and finishing time of each vertex, and show the classification of each edge.

9. (2 pts) Show the ordering of vertices produced by TOPOLOGICAL-SORT when it is run on the graph shown in Figure 4. Assume that the for loop of DFS procedure considers the vertices in alphabetical order and that the adjacency-lists are ordered by alphabetical order.

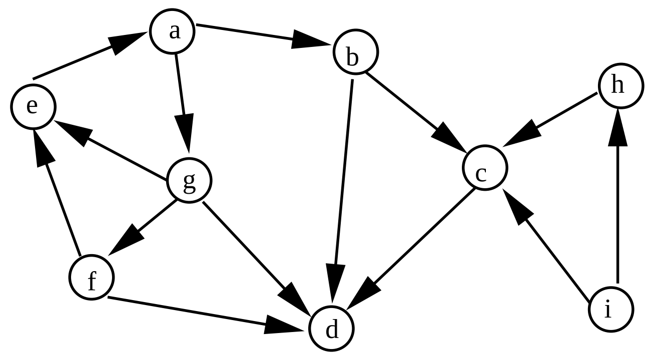

Figure 3: DFS

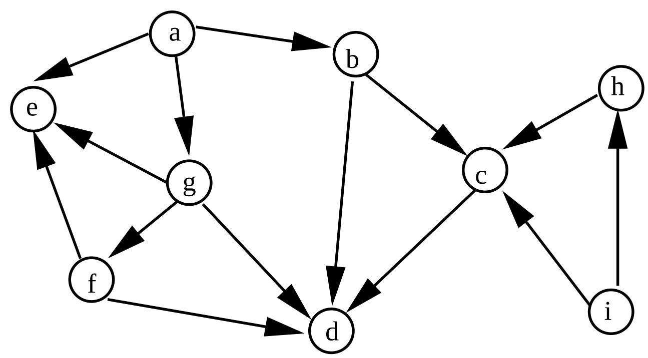

Figure 4: Topological Sort.

10. (7 pts) Page 614, 22.4-2.

11. (6 pts) Consider an undirected connected graph $G = (V, E)$. If we run the BFS algorithm on G, a BFS-tree T_b results. If we run the DFS algorithm on G, a DFS-tree T_d results. If $T_b = T_d$, what can you say about G. Justify your answer.