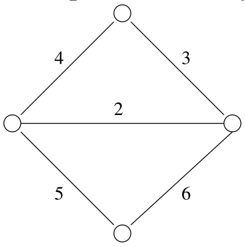
CSE548, Fall 2004

Solutions/Hints to Homework 5

23-1 (a) The minimum spanning tree (MST) must be unique. This can be verified by following Kruscal's algorithm. In Kruscal's algorithm, each step it chooses a unique edge to cosider because all weights are distinct.

The second-best MST is not unique as shown by the following example:



In the above graph, the best MST consists of edges of weights $2, 3, 5$. There are two second-best MSTs, one having edges of weights $2, 4, 5$ and the other one having edges of weights $2, 3, 6$.

(b) Let T' be a second best MST. Let $\{u, v\} \in T - T'$. Then, $T' \cup \{u, v\}$ contains a cycle and one of the edges in the cycle is not in T. Let this edge by $\{x, y\}$. Then, we must have $w(x, y) > w(u, v)$, for otherwise, we could replace $\{u, v\}$ in T by $\{x, y\}$ to get a MST better than T. Now, we note that $T'' = T' - \{x, y\} \cup \{u, v\}$ is also a spanning tree since $\{u, v\}$ and $\{x, y\}$ are in the same cycle. In addition, $w(T'') < w(T')$. So, T'' is a best MST. By the uniqueness of MST of part (a), we see that $T'' = T$ and T and T' differ with only one edge.

(c) The main observation is that the path between two vertices u and v in the MST T is unique. The algorithm is as follows:

> (1) Find the MST T, and adjust the Adacency List of the graph to include only edges in T.
>
> (2) For each $u \in V$, perform a BFS or DFS to find the maximum weight between u and every other $v \in V$. Since T is a spanning tree, the BFS Tree or DFS Tree has only tree edges. When we visit edge (x, y) (from x to y), the max-weight edge in the path from root u to y is found (either the max-weight edge from u to x or the edge (x, y), depending upon which weight is higher).

We note that step (1) takes time $O(E+V \lg V) = O(V^2)$ (by Prim's algorithm with the implementation by Fibonacci heap) and, for each u, the time for BFS or DFS is $O(V)$ because the tree T has only $V - 1$ edges. So, the total time is $O(V^2) + O(V^2) = O(V^2)$.)

(d) From (b), we know that we can replace one edge $\{u, v\}$ in the bst MST T by a new edge $\{x, y\}$ to get a second-best MST T'. How do we determine these two edges? Note that if we replace $\{u, v\}$ by $\{x, y\}$, then $w(T') = w(T) - w(u, v) + w(x, y)$. If we know which $\{x, y\}$ to add to T', then $\{u, v\}$ must be the max-weight edge in the path from x to y, which can be found by part (c). So, we get the following algorithm:

(1) Find MST T.

(2) Find max-weight edges as in part (c). Let $max(x, y)$ denote the max-weight edge in the path from x to y in tree T.

(3) Find an edge $\{x, y\} \in E - T$ that minimizes $w(x, y) - w(max(x, y))$.

(4) Output $T' = T - max(x, y) \cup \{x, y\}$.

We have seen, in part (c), that steps (1) and (2) can be done in time $O(V^2)$. There are $E = O(V^2)$ edges to be considered in step (3), and so step (3) takes time $O(V^2)$. The total runtime is $O(V^2)$.

24-1 (a) Obvious. In G_f, all edges (e_i, e_j) have $i < j$ and so there cannot be a cycle.

(b) Without loss of generality, assume that $s = v_1$. For any vertex v_k, consider the shortest path π from s to v_k: $\pi = (v_{i_1}, v_{i_2}, \ldots, v_{i_m})$, where $i_1 = 1$ and $i_m = k$. Let $A = \{v_{i_j} \mid 1 \le j \le m, d[v_{i_j}] = \delta(s, v_{i_j})\}$. We claim that each pass of Yen's algorithm adds at least two vertices to set A (unless $|A| \ge m - 1$). More precisely, let π_1 be the subpath of π that starts at s and cons ists of a maximal sequence of edges in E_f, followed by a maximal sequence of edges in E_b. That is, let $t_1 = \max\{j \mid j \le m, i_1 < i_2 < \cdots < i_j\}$ and $t_2 = \max\{j \mid j \le m, i_{t_1} > i_{t_1+1} > \cdots i_j\}$, and $\pi_1 = (v_{i_1}, v_{i_2}, \cdots, v_{i_{t_2}})$. Then, we can see that in the first pass of Yen's algorithm, the relaxation follows the order of (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), $\ldots.$ $(v_{i_{t_1-1}}, v_{i_{t_1}})$ in E_f, and then $(v_{i_{t_1}}, v_{i_{t_1+1}})$, $(v_{i_{t_1+1}}, v_{i_{t_1+2}})$, $\ldots.$ $(v_{i_{t_2-1}}, v_{i_{t_2}})$ in E_b. So, by the convergence theorem, all these vertices $v_{i_1}, \ldots, v_{i_{t_2}}$ are in set A. Note that $i_1 < i_2$ implies that $t_1 > i_1$ and $t_2 < t_1$, and so we added, in this pass, at least two vertices to A.

A similar argument shows that, in each later pass, we add at least two more vertices to A unless $|A| \ge m - 1$. Since $m \le |V|$, we need at most $\lceil |V|/2 \rceil$ passes to get $v_k \in A$.

25.1-5 Assume that $V = \{1, 2, \ldots, n\}$ and $s = 1$. Let $Mn \times n$ be the weight matrix corresponding to graph G; i.e., $M[i, j] = w(i, j)$, for $i, j \in \{1, 2, \ldots, n\}$. Let a be the $1 \times n$ vector $[0, \infty, \ldots, \infty]$. Then, the vector $b = a \cdot M^{(n)}$ is the