Solutions for Chapter 23

Zhixiang Zhu
zzxiang21cn@hotmail.com

February 15, 2013

Solution to Exercise 23.1-2

Let $V = \{a, b, c\}$, $E = \{(a, b), (b, c), (a, c)\}$, $w(a, b) = 1$, $w(b, c) = 2$, $w(a, c) = 3$, $A = \emptyset$ and $S = \{b\}$. (b, c) is a safe edge for A, but it is not a light edge for the cut.

Solution to Exercise 23.1-3

If we remove (u, v) from the minimum spanning tree it is in, the spanning tree will be disconnected and divided into two components. Let one of the two components be S, then (u, v) must be a light edge crossing the cut $(S, V - S)$; otherwise we can add a light edge of the cut to the disconnected tree so that the newly formed spanning tree has a smaller total weight, contradicting that (u, v) is in a minimum spanning tree.

Solution to Exercise 23.1-4

Let's say we have a graph consisting of three vertices: u, v, w; and the weights of (u, v), (v, w) and (u, w) are all equal.

Solution to Exercise 23.1-5

Because e is on a cycle, any cut crossed by e must also be crossed by some other edge on the cycle. If we build the minimum spanning tree by choosing light edges, e will never be chosen, so that the constructed minimum spanning tree will not include e.

Solution to Exercise 23.1-7

If all edge weights are positive, such a subset of edges must be acyclic; otherwise we can remove the edges who have bigger weights until the subset becomes acyclic to get a smaller total weight. Therefore, it must be a tree. If we allow some weights to be nonpositive, this would not hold. Example: suppose that a graph contains a cycle on which all the edge weights are negative, then such a subset must contains the cycle.

Solution to Exercise 23.1-8

Consider any edge $(u, v) \in T$. If we remove (u, v) from T, T will be disconnected, resulting in a cut $(S, V - S)$. The edge (u, v) is a light edge crossing the cut $(S, V - S)$ (by Exercise 23.1-3). Now consider the edge $(x, y) \in T'$ that crosses $(S, V - S)$. It, too, is a light edge crossing this cut, so that the weights of the edges (u, v) and (x, y) are equal. Therefore, for every edge (u, v) in T, there must be an edge (x, y) in T' whose weight is equal to that of (u, v), so that the list L is also the sorted list of edge weights of T'.

Prof. Ming-Deh Huang has another solution, see `Prof Ming-Deh Huang.pdf` in directory `Unofficial Solutions to Homework`.

Solution to Exercise 23.1-9

If there's a spanning tree T'' of G' whose total weight is smaller than that of T', we can substitute T' with T'' in T to get a spanning tree of G, whose total weight is smaller than that of T, so that T is not a minimum spanning tree of G, which is a contradiction.

Solution to Exercise 23.1-10

If the weight of edge (x, y) is decreased by k, the total edge weight W of T will also be decreased by k, so that every minimum spanning tree for G with edge weights given by w' should have its total edge weight W' not larger than $W - k$. We now show that W' is not smaller than $W - k$. Every minimum spanning tree for G with edge weights given by w' must contain the edge (x, y); otherwise because other edge weights have not changed, its total edge weight should be not smaller than W. For any of these minimum spanning trees, because the only edge who has its weight changed is (x, y), its total edge weight must be not larger than $W - k$. Therefore, the total edge weight of any minimum spanning tree for G with edge weights given by w' is $W - k$, and T is still such a minimum spanning tree.

Solution to Exercise 23.1-11

Let (u, v) be the edge not in T whose weight is decreased. Using DFS or BFS, find the unique simple path from u to v in T. Find an edge e of maximal weight on that path. If the weight of e is greater than that of (u, v), replace e in T with (u, v).

Solution to Exercise 23.2-1

First sort the edges in T, and then insert the remaining edges into the sorted list. During the insertion of each remaining edge e, make sure it is inserted after those edges already in the list whose weights are also $w(e)$.

Solution to Exercise 23.2-2

The only place in MST-PRIM that makes difference between adjacency lists and adjacency matrix is line 8. If we apply an adjacency matrix implementation directly to MST-PRIM, the running time would become $O(V^2 \lg V)$.

Let's denote the elements of the adjacency matrix by a_{ij}.

MST-PRIM-ADJACENCY-MATRIX(G)

```
for i ← 1 to |V|
     do key[i] ← ∞
        π[i] ← NIL
        included[i] ← FALSE
key[1] ← 0
included[1] ← TRUE
n ← 1
i ← 1
while n < |V|
     do for j ← 1 to |V|
             do if included[j] = FALSE and a_ij < key[j]
                  then π[j] ← i
                       key[j] ← a_ij
        k ← 1
        for j ← 1 to |V|
             do if included[j] = FALSE and key[j] < key[k]
                  then k ← j
        included[k] = TRUE
        i ← k
        n ← n + 1
```