Introduction To Algorithm
Third Edition
Answer

Xia Ding

December 19, 2015

23.1

23.1–1

Theorem 23.1 in book shows this. Let A be the empty set and S be any set containing u but not v.

23.1–2

Suppose we have a graph G with many edges, there is a node with which only one edge e is incident. e must be a safe edge for A, but it may be not a light edge for some cut if it has a very high weight.

23.1–3

Let (S, V−S) be a cut such that $u \in S$ and $v \in V - S$, because (u, v) is contained in some minimum spanning tree, so it is a light edge crossing this cut.

23.1–4

A triangle whose edge weights are all equal is a graph in which every edge is a light edge crossing some cut. But the triangle is cyclic, so it's not a minimum spanning tree.

23.1–5

Because e is in some cycle, every nodes in that cycle should be in some minimum spanning tree T. Let that cycle be a cycle in which no other cycle exist. So some edge in that cycle can't be in T, marked it as e_1. If $e_1 = e$, we are done, if not, suppose e is in some minimum spanning tree T_1, then we can construct a T_2 such that $w(T_2) \leq w(T_1)$ by removing e and add e_1 because $w(e_1) \leq w(e)$. So there is a minimum spanning tree of G that doesn't include e.

23.1–6

Suppose that for every cut of G, there is a unique light edge crossing the cut . Let us consider two minimum spanning trees, T and T', of G. We will show that every edge of T is also in T', which means that T and T' are the same tree and hence there is a unique minimum spanning tree.
Consider any edge $(u, v) \in T$. If we remove (u, v) from T, then T becomes 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 cross $(S, V - S)$ is unique, the edges (u, v) are the same edge. Thus, $(u, v) \in T'$. Since we chose (u, v) arbitrarily, every edge in T is also in T'.

A counter example is such G that $V = x, y, z$, $E = (x, y, 1), (x, z, 1)((a, b, w)$ *is a edge* (a, b) *weighted* $w)$. Consider the cut (x, y, z). Both of the edges (x, y) and (x, z) are light edges crossing the cut, and they are both light edges.

23.1–7

According to the definition, if all edge weights are positive, then we can't have a cycle in minimum spanning graph, because if exist, we can always remove an edge without removing nodes from the minimum spanning graph to decrease the total weight. So any subset of edges that connects all vertices and has minimum total weight must be a tree.

If we allow some weights to be non-positive, then we can construct a cycle without increasing the total weight if that cycle contains edges weighted non-positive, so that some subset of edges that connects all vertices and has minimum total weight can't be a tree.

23.1–8

23.1–9

If T' isn't a minimum spanning tree of G', then there exists a path $ve_1e_2 \ldots e_nu$ to replace original (v, u) in T' so that total weightof T' will decrease. Because T' is a subset of T, so (v, u) is in T too, then we can use $ve_1e_2 \ldots e_nu$ to replace (v, u) in T so that total weight of T will decrease, contradicting that T is a minimum spanning tree.

23.1–10

Suppose we make initial cut as $(, V-)$, then after we construct a MST with m edges, then they are m minimum weight edges of G and (x, y) is in them. After decrease (x, y)'s weight, those edges are still m minimum weight edges in G, so T is still a MST of G.

23.1–11

23.2

23.2–1

23.2–2

Algorithm ..

Algorithm 1 MATRIX-PRIM(G, w, r)

for all $u \in G.V$ **do**
    $u.key = \infty$
    $u.\pi =$ NIL
end for
$r.key = 0$
$Q = G.V$
while $Q \neq \emptyset$ **do**
    $u =$ EXTRACT-MIN(Q)
    for $v = 1$ to $|V|$ **do**
        if $G.A[u, v] \neq 0$ **and** $j \in Q$ **and** $w(u, v) < v.key$ **then**
            $v.\pi = u$
            $v.key = w(u, v)$
        end if
    end for
end while

23.2–3

23.2–4

23.2–5

23.2–6

23.2–7

23.2–8