Homework 3

Obliviate

March 2021

1 Solution:

Assume that all the different values of weight are $w_1, w_2, \cdots, w_k$ with $w_i < w_{i+1}$ for all i. Specially we define $w_0 = -\infty$.

Suppose that there exists c so that T_c and G_c have different connected components, which implies that there is an edge (x, y) which weighs no more than c so that x and y are not in the same connected component of T_c while they are obviously in the same connected component of G_c.

Let i be the maximum integer so that $w_i \leq c$, it's easy to see that $T_{w_i} = T_c$ and $G_{w_i} = G_c$. And let j be the minimum integer so that x and y are in the same connected component in T_{w_j}, which must be greater than i. And denote (u, v) to be $T_{w_j} \backslash T_{w_{j-1}}$.

Then $T \backslash \{(u, v)\} \cup \{(x, y)\}$ is also a spanning tree of G. But since $w_j > w_i$, the weight of new spanning tree is smaller than T, which leads to contradiction.

2 Solution:

We use $n(G)$ to denote the number of vertices in G, $c(G)$ to denote the number of connected components in G, $m(G)$ to denote the number of edges in G.

When T is a tree, we have $m(T) = n(T) - 1$. When T is a forest, we have $m(T) = n(T) - c(T)$.

$m_c(T) = m(T_c) = n(T_c) - c(T_c) = n(G) - c(T_c)$.

$m_c(T') = m(T'_c) = n(T'_c) - c(T'_c) = n(G) - c(T'_c)$.

We have known that T_c, T'_c, G_c have the same connected components, so $c(T_c) = c(T'_c)$.

And then $m_c(T) = m_c(T')$.

3 Solution:

Assume that we have two different MST T_a and T_b. The weight of edges of the two trees are $a_1, a_2, ..., a_k$ and $b_1, b_2, ..., b_k$ sorted in ascending order. So there exists the smallest i that $a_i \neq b_i$. Assume that $a_i < b_i$. Inserting the edge weighted a_i to T_b will form a cycle, and there exists at least one edge weighted larger than a_i in the cycle, which means that T_b is not MST. So MST is unique.

4 Solution:

The middle edge must exist.

The left part has 7 spanning trees.

The right part has 7 spanning trees.

So the graph has $7 \times 7 = 49$ spanning trees.

5 Solution:

We denote all the weight as $w_1, w_2, ..., w_k$ with $w_i < w_{i+1}$ for all i. Then we partition edges into disjoint sets $E_1, E_2, ..., E_k$ with $E_i = \{e \in E(G) | w(e) = w_i\}$.

The algorithm has k steps in total. We start with $G' = (V(G), \emptyset)$. In the i-th step, we insert all the edges in E_i to G', and calculate the number of spanning forests of current G' which can be done by multiplying the numbers of spanning trees in all connected components. We denote the answer of this step to be a_i. Then for every connected component in G', we remove all the vertices in the component, and create a new vertex named u, and relabel all the vertices in the component to u. After relabelling the vertices in this step, we have an updated G' with no edges again. Repeating the step, we could get $a_1, a_2, ..., a_k$. The number of minimum spanning trees of G is $\prod\limits_{i=1}^{k} a_i$.

The correctness is supported by the conclusion in Ex1.