Homework #3 **Coffee Automaton**
CS 217 @ SJTU ACM Class, Zhiyuan College, SJTU
Prof. **Dominik Scheder** Due Date: September 30, 2019
TA. **Tang Shuyang** Submit Date: September 30, 2019

Exercise 1

It's trivial that T is a subgraph of G, and T_c is a subgraph of G_c.

So for two vertices u, v, if u, v are connected in T_c, they must be connected in G_c, because T_c is a subgraph of G_c.

If u, v are not connected in T_c, they are not connected in G_c. A simple proof: because T is a minimum spanning tree of G, all vertices are connected in T. If u, v are not connected in T_c, then there is at least an edge connecting u and v which has a weight greater than c. Assuming u, v are connected in G_c, then there is a path connecting u and v such that every edge weight is less or equal to c. So we can find a tree that is a spanning tree of G, which has a weight less than T. But T is the MST, so it is a contradiction.

So T_c and G_c have exactly the same connected components.

Exercise 2

Considering the Kruskal's algorithm, all MSTs have $n-1$ edges, and have same number of edges with weight 1, weight 2, ... , weight c. Therefore, all T_c have the same number of edges, namely $m_c(T) = m_c(T')$.

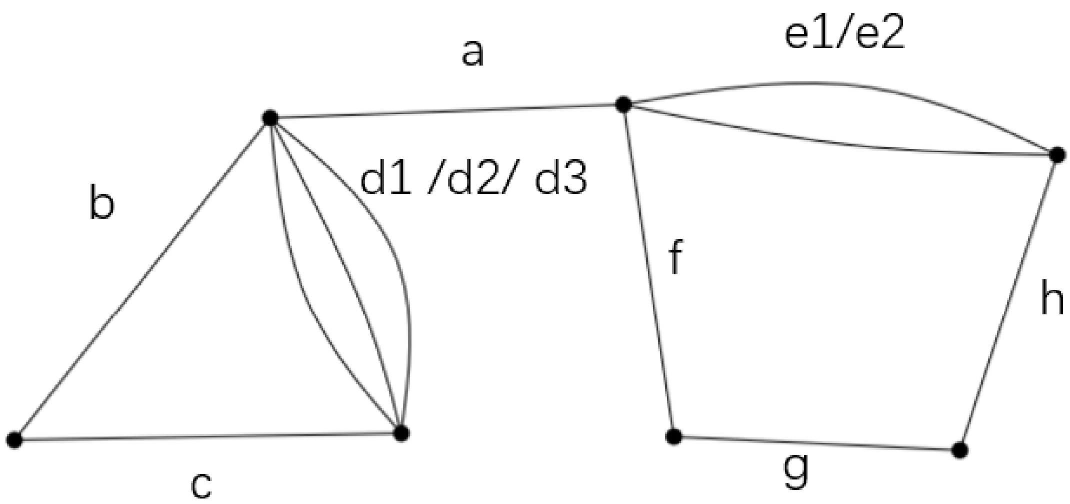
Exercise 3

Considering the Kruskal's algorithm, each time we choose edges of fixed weight and loop $n-1$ times. Since no two of edges have the same weight, what we choose is deterministic and unique. Therefore, there is exactly one MST.

Exercise 4

The multigraph is showed in figure with labeled.

Since we want a spanning tree, edge a is required, and choose the two edges of b, c, d, the three edges of e, f, g, h. And we have three choices in d and two choices in e, so we have $(2 \times 3 + 1) \times (3 \times 2 + 1) = 49$ spanning trees.



Exercise 5

Let T be a minimum spanning tree of G.

By exercise 2, we know that the connectivity of T_c is uniquely determined.

The overall algorithm can be obtained by making a little modification to the Kruskal's Algorithm.

1. Sort all edges in the graph.
2. Traverse the sorted edges and group them together for the same edge weight w.
3. Calculate the number of spanning trees with the weight of w, and multiply the answer by this number. Then go to step 2.