1 Let $G = (V, E)$ be the original graph. Suppose there are two distinct MSTs $T_1 = (V, E_1)$ and $T_2 = (V, E_2)$. The sets $(E_1 - E_2)$ and $(E_2 - E_1)$ are not empty, so there are at least one e belongs to $(E_1 - E_2)$. Since e does not belong to E_2, adding it to T_2 creates a cycle. By cycle property, the edge e' that has the most weight of this cycle does not belong to any MST. If $e' = e$, then $e' \in E_1$ (because $e \in E_1 - E_2$); if $e' \neq e$, then $e' \in E_2$. Now both cases are contradicting to the fact that e' is not in any MST. Hence, G has exactly one minimum spanning tree T.

2 Given any cut in a edge-weighted graph, one of the crossing edges of minimum weight is in the MST of the graph.

Proof:
Let $e.f$ be one of the crossing edges of minimum weight and T be the MST. Suppose that T does not contain $e.f$. Now consider the graph formed by adding $e.f$ to T. This graph has a cycle that contains $e.f$, and that cycle must contain at least one crossing - say, g, which has heavier weight than e. We can get a spanning tree of strickly lower weight by deleting g and adding e or f, contradicting the assumed minimality of T.

3 There are six MSTs in total (weight) (profile)

T_1 : E_1 = {(b,c), (c,d), (d,a), (b,e)}, W=7, 1, 1, 2, 3

T_2 : E_2 = {(b,c), (c,d), (d,a), (c,e)}, W=7, 1, 1, 2, 3

T_3 : E_3 = {(b,c), (c,d), (d,a), (d,e)}, W=7, 1, 1, 2, 3

T_4 : E_4 = {(a,b), (b,c), (d,a), (b,e)}, W=7, 1, 1, 2, 3

T_5 : E_5 = {(a,b), (b,c), (d,a), (c,e)}, W=7, 1, 1, 2, 3

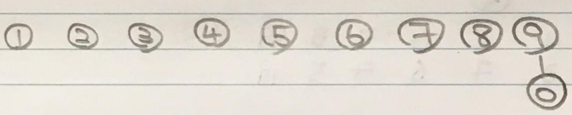
T_6 : E_6 = {(a,b), (b,c), (d,a), (d,e)}, W=7, 1, 1, 2, 3

The profiles of the MSTs of G are all the same

4

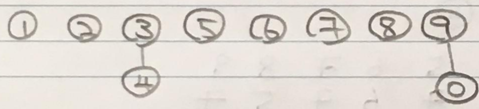
index	0	1	2	3	4	5	6	7	8	9
id[]	0	1	2	3	4	5	6	7	8	9

(9-0)



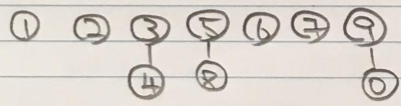
index	0	1	2	3	4	5	6	7	8	9
id[]	9	1	2	3	4	5	6	7	8	9

(3,4)



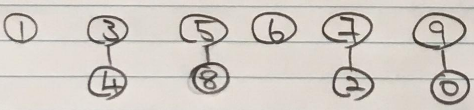
index	0	1	2	3	4	5	6	7	8	9
id[]	9	1	2	3	3	5	6	7	8	9

(5-8)



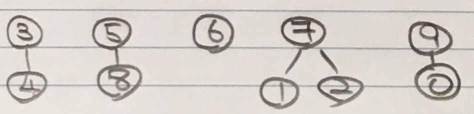
index	0	1	2	3	4	5	6	7	8	9
id[]	9	1	2	3	3	5	6	7	5	9

(7-2)



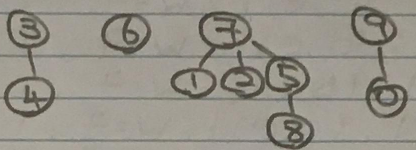
index	0	1	2	3	4	5	6	7	8	9
id[]	9	1	7	3	3	5	6	7	5	9

(2-1)



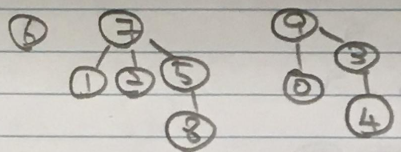
index	0	1	2	3	4	5	6	7	8	9
id[]	9	7	7	3	3	5	6	7	5	9

(5-7)



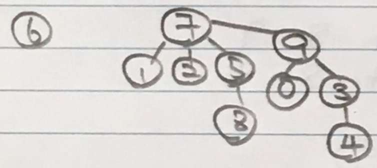
index	0	1	2	3	4	5	6	7	8	9
id[]	9	7	7	3	3	7	6	7	5	9

(0,3)



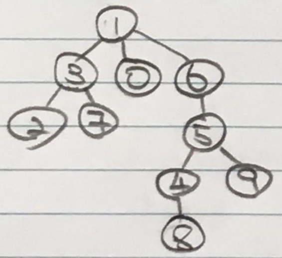
index	0	1	2	3	4	5	6	7	8	9
id[]	9	7	7	9	3	7	6	7	5	9

(4-2)



index	0	1	2	3	4	5	6	7	8	9
id[]	9	7	7	9	3	7	6	7	5	7

5



It's impossible because max depth is 5, which is greater than log 10.