CS208 · Algorithm Design and Analysis

何泽安 (He Zean)
12011323

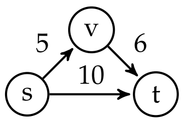
Assignment 4

Mar. 30, 2022

Ch.4 - Ex.2

(a) True It's trivial that for any two different positive number c_1 and c_2, if $c_1 < c_2$ then $c_1 \cdot c_1 < c_2 \cdot c_1 < c_2 \cdot c_2$, aka. $c_1^2 < c_2^2$, vice versa. Assume that we are using Kruskal algorithm to generate the MST, which first sort the edges by their weight increasingly, then each term insert the edge with least weight, if this would not create a circle, otherwise, give up this edge and choose the next one, until all edges are considered. Assume that the original edges are sorted by weight as $c_{e1} < c_{e2} < \cdots < c_{en}$, by the proof above we have $c_{e1}^2 < c_{e2}^2 < \cdots < c_{en}^2$. Since the vertexes each edge connects don't change, which means the order of constructing a subgraph keeps the same, aka. the order of accepting / rejecting each edge is the same of the previous case. Finally, Kruskal algorithm handles the edges in the original order, and give the original judge, eventually choosing the same edges as before (with the same edges and vertexes, this MST is very same as T).

(b) False Consider such counter-example:



It's trivial that the minimum-cost s-t path is $\{s - t\}$, costing 10, but after squaring the edges' cost, the minimum-cost s-t path changes to $\{s - v - t\}$, costing 61.

Ch.4 - Ex.8

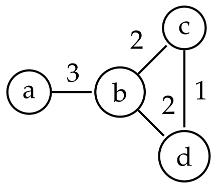
Suppose that G's MST is not unique, say, we can find (at least) two different trees T_1 and T_2, having the same minimum cost. By the definition of MST, T_1 and T_2 must have the same vertexes, but by the definition of tree, different tree must have different vertex or edge, thus, we conclude there must be at least one edge e that in either T_1 or T_2, but not in both (here we let e in T_1, actually vice versa).

Now we can add e into T_2, then by the definition of MST, there must be a cycle in T_2 now. In the set of edges constructing this cycle, we take the edge with the maximum cost as e_m. By the property of cycle, we know that e_m should no way be involved in a MST. ① If e_m is e, then T_1 involves an edge (having the largest weight in a cycle) that should not be in MST; ② If e_m is not e, T_2 itself involved an edge should not in MST. Both cases break one "MST", leaving another, which contradicts to our assumption that T_1 and T_2 are both MST.

The assumption that G's MST is not unique is therefore been disproved, aka. a connected graph with distinct costs of edges has a unique MST.

Ch.4 - Ex.22

T needs not to be a MST. Consider such counter-example:



This graph have two MSTs, $\text{MST}_1 = \{a-b, b-c, c-d\}$ and $\text{MST}_2 = \{a-b, b-d, c-d\}$. Now consider $T = \{a-b, b-c, b-d\}$, having $T \subseteq E$ and $(\forall e)(e \in T \rightarrow e \in \text{MST})$, but it's also trivial that T is not a MST.