Assignment 3
CSCI 6101

B00755354
Yansong Li

Question 1 (Reduction to a Related Problem)

Part (a)

In order to prove T is a Steiner tree of G, we need to prove it includes all the vertices in S.

Algorithm of how to get T: ① G → ② H → ③ T' → ④ G' → ⑤ T. What we need to do is proving that each stage of the algorithm includes all vertices in S.

Proving ① Since S is a given subset of vertices $S \subseteq V$, and V is the set of vertices in G. Hence, G must includes all the vertices in S.

Proving ② Since H is an auxiliary graph that constructed with vertex set S, it must includes all the vertices in S.

Proving ③ Since T' is a minimum spanning tree of H and H includes all the vertices in S from ②. Also, the definition of a spanning tree of original graph is it contains all the vertices in the graph. Hence, T' contains all the vertices in H. Therefore, T' must contains all the vertices in S.

Proving ④ Since H is constructed with vertex set S, and each edge $(u,v) \in H$ is a representation of a shortest path $P_{u,v}$ from u to v in G with the length of $P_{u,v}$ as its weight. Also, T' is a minimum spanning tree of H. Therefore, all the edges in T' is a subset of edges in H and all of the edges represent a set of shortest paths in G. For constructing the subgraph $G' \subseteq G$, an edge is contained if and only if this edge belongs to some path $P_{u,v}$ such that $(u,v) \in T'$. Since all the edge in T' represent some path $P_{u,v}$ in G, G' is only replacing each edge in T' by its corresponding shortest path in G. Therefore, $|V'_{T'}| \leq |V'_{G'}|$, which means the number of vertices in T' is no more than the number of vertices in G' (extra vertices in G' are all non-terminals). All vertices in S are reserved during the tranformation from T' to G'. Hence, G' contains all the vertices in S.

Proving

③ since T' is an arbitrary spanning tree of G' and a spanning tree contains all the vertices of the original graph. Hence, T contains all the vertices in G', it must contains all the vertices in S as well.

As a result, we have proved that T is a Steiner tree of G.

Part (b)

In order to prove T is a 2-approximation of a minimum Steiner tree T^*, we need to prove that

① the weight of the final tree T returned by the algorithm is no more than the weight of T', $w(T) \leq w(T')$.

② the weight of T' is no more than the weight of an Euler tour of a minimum Steiner tree T^*, $w(T') \leq 2w(T^*)$

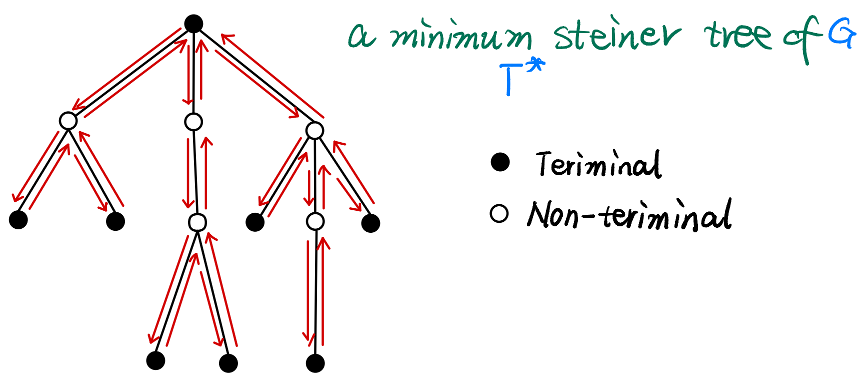
Proving

① . Since all the edges in T' represent some shortest path $P_{u,v}$ in G, constructing G' is just replacing each edge in T' by its corresponding shortest path in G. Therefore, the total weight of edges that belong to some path from one terminal u to another terminal v in G' equals to the weight of the corresponding edge (u,v) in T'. Also, since paths between different terminals in G' could share some common edges, the total weight of G' is no more than the total weight of T', $w(G') \leq w(T')$. As T is an arbitary spanning tree of G', its weight is no more than the weight of G', $w(T) \leq w(G')$. Hence, the weight of T is no more than the weight of the MST T' of H, $w(T) \leq w(T')$.

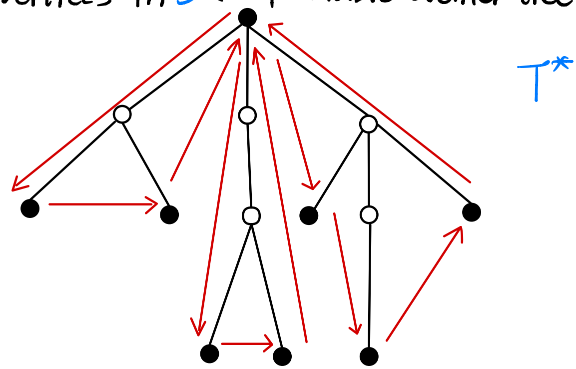
Proving

② Let T^* be a minimum steiner tree of G. By conducting an Euler tour P with depth-first search, each edge in T^* has been visited twice. Hence, total weight of the edges in P is twice the weight of T^*, $w(P) = 2w(T^*)$.

Example:



Next, traversing the Euler tour P and skipping the non-terminal vertices and traversed terminal vertices, which forms a Hamiltonian cycle A that covers all vertices in S (a feasiable steiner tree, but not optimal).



Base on the triangle inequality, that is the length of an edge c in a triangle is no more than the sum of the lengths of the other two edges a and b, $c \leq a+b$. Therefore, the weight of this Hamiltonian cycle A is no more than the weight of the Euler tour P, which means it is also no more than twice of the weight of T^*, $w(A) \leq w(P) = 2w(T^*)$.

Next, we delete an arbitary edge from the Hamiltonian cycle A to obtain a spanning tree of the vertices in S. Since this spanning tree is one edge less than A, the weight of it is no more than the weight of A. Meanwhile, T' is a minimum spanning tree that covers all the vertices in S; hence, the weight of T' is no more than the weight of the spanning tree from A. Therefore, the weight of T' is no more than the weight of A as well. Hence, the weight of T' is no more than the weight of the Euler tour P that was conducted on T^*. Hence, the weight of T' is no more than the twice of the weight of T^*, $w(T') \leq w(A) \leq w(P) = 2w(T^*)$.

As a result, $w(T) \le w(T')$ and $w(T') \le w(A) \le w(P) = 2w(T^*)$. Hence, the weight of the final tree T returned by the algorithm is no more than the twice of the weight of the minimum steiner tree T^*, $w(T) \le 2w(T^*)$.

Hence, T is a 2-approximation of a minimum steiner tree.