CS602 Applied Algorithms **Spring 2021**

Homework - Steiner Tree/Forest (Mar 11) No submission

Q1. Let $G(V, E)$ be a graph and let $s_1, t_1, s_2, t_2 \ldots, s_p, t_p \in V$ be designated as terminal vertices. Let $F \subseteq E$ be a subset of edges such that the following holds for every cut $S \subset V$

$$\text{if for some } 1 \leq j \leq p, s_j \in S \text{ but } t_j \notin S \text{ then } F \cap \delta(S) \geq 1.$$

Prove that for every $1 \leq j \leq p$, there is a path between s_j and t_j in the subgraph (V, F).

Q2. In the primal-dual algorithm discussed for Steiner Forest, if we skip the pruning step in the end, will our analysis still show that the algorithm has a 2-approximation guarantee.

Q3. Consider the Steiner tree problem where the set of terminals is the set of all vertices, i.e., we want a minimum weight spanning tree. Will the primal-dual algorithm discussed always give the minimum weight spanning tree? Does the algorithm look similar to Kruskal's algorithm in this case? Is it exactly the same?

Q4. Consider the Steiner tree problem when we have only two terminals s and t, i.e., we are looking for a shortest path between s and t. Can you show that the primal-dual algorithm discussed has approximation factor 1 in this case. That is, it will always give the shortest path.

In the primal-dual algorithm here, we will maintain two active components, one around s and one around t. Suppose instead, we just maintain just one active component which is around s and keep growing it by changing the dual variables appropriately. Then does this algorithm look similar to Dijkstra's algorithm.

Q5 (Survivable Network Desgin). In this problem, we are given pairs of vertices along with numbers $(s_1, t_1, n_1), (s_2, t_2, n_2), \ldots, (s_k, t_k, n_k)$ and want to find out the minimum weight subgraph which has at least n_i edge-disjoint paths between s_i and t_i, for each i. Design an approximation algorithm for this problem using a primal dual scheme similar to the Steiner Tree/Forest problem. Is it a 2-approx algorithm?

Q6 It should be a good programming exercise to implement the primal dual Steiner Tree/Forest algorithm.