Approximation Algorithms	**Dept. of CS, Aarhus University**
Homework 2	
Lecturer: Nguyen Kim Thang	*Student: ??*

1 Problems

The set of problems is 3.2, 3.3, 4.2 in the Approximation Algorithms book. You choose to solve one of them and solve the following question in class.

Class problem

Consider the following greedy algorithm for STEINER TREE. Given a graph $G(V, E)$, cost on edges $c : E \rightarrow \mathbb{Q}^+$ and a set of terminals S.

Algorithm 1 A greedy algorithm for STEINER TREE

Let $\{s_1, \ldots, s_k\}$ be an arbitrary order of vertices in S where $k = |S|$.
$T \leftarrow \emptyset$
for $i = 1$ to k **do**
    Let P_i be a shortest path from s_i to T.
    $T \leftarrow T \cup P_i$.
end for
return T.

Prove that the greedy algorithm gives $O(\log k)$-approximation.

Hint: Let $c(P_i)$ be the cost of the path P_i. Let $\{i_1, \ldots, i_k\}$ be a permutation of {1, ..., k} such that $c(P_1) \geq \ldots \geq c(P_k)$. The proof is done by two steps:

1. Prove that $c(P_{i_j}) \leq 2OPT/j$ where OPT is the cost of an optimal Steiner tree on terminals S.

   Prove by contradiction. Let j be the smallest index such that $c(P_{i_j}) > 2OPT/j$. Consider the set $\{s_{i_1}, \ldots, s_{i_j}\}$ and think about minimum spanning tree.

2. Using the first step to prove $O(\log k)$-approximation.

Note: this is an *online* algorithm: terminals are considered in an arbitrary order, and when a terminal is considered, it is immediately connected to the existing tree. Thus, even if the algorithm could not see the entire input at once, but instead terminals were revealed one at a time and the algorithm had to produce a Steiner tree at each stage, Algorithm 1 outputs a tree of cost no more than $O(\log |S|)$ times the cost of the optimal tree.

Proof of the class problem.

1. Suppose by contradiction this were not true; since s_{i_j} is the terminal with j^{th} highest cost of connection, there must be j terminals that each pay more than $2OPT/j$ to connect to the tree that exists when they are considered. Let $S' = \{s_{i_1}, s_{i_2}, \ldots s_{i_j}\}$ denote this set of terminals.

We argue that no two terminals in $S' \cup \{s_1\}$ are within distance $2OPT/j$ of each other. If some pair x, y were within this distance, one of these terminals (say y) must be considered later by the algorithm than the other. But then the cost of connecting y to the already existing tree (which includes x) must be at most $2OPT/j$, and we have a contradiction.

Therefore, the minimum distance between any two terminals in $S' \cup \{s_1\}$ must be greater than $2OPT/j$. Since there must be j edges in any MST of these terminals, an MST must have cost greater than $2OPT$. But the MST of a subset of terminals cannot have cost more than $2OPT$, as argued in the class. Therefore, we obtain a contradiction.

2. Given the first step, we have:

$$\sum_{i=1}^{|S|} c(P_i) = \sum_{j=1}^{|S|} c(P_{i_j}) \leq \sum_{j=1}^{|S|} \frac{2OPT}{j} = 2OPT \sum_{j=1}^{|S|} \frac{1}{j} = 2H_{|S|} \cdot OPT.$$

$\square$