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 $c : E \to \mathbb{Q}^+$ and a set of terminals S.

Algorithm 1 A greedy algorithm for STEINER TREE

```
Let {s_1, ..., s_k} be an arbitrary order of vertices in S where k = |S|.
T ← ∅
for i = 1 to k do
    Let P_i be a shortest path from s_i to T.
    T ← T ∪ 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.