Algorithmic Discrete Mathematics 5. Exercise Sheet

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Mathematics
PD Dr. Ulf Lorenz
Dipl.-Math. David Meffert

SS 2012
13. and 14. Juni 2012
Version of June 11, 2012

Groupwork

Exercise G1 (Spanning trees I)
Let $G = (V, E)$ be a connected, undirected, weighted graph with weight function $w\colon E \to \mathbb{N}$. Prove the following statements about spanning trees:

(a) Any edge $e \in E$ that is not contained in any cycle of G is contained in every minimal spanning tree of G.

(b) Every graph G with pairwise disjoint weights on the edges contains a unique minimal spanning tree.

Exercise G2 (Spanning trees II)
Let $G = (V, E)$ be a undirected, connected, weighted graph.

(a) Let (u, v) be an arbitrary edge of a minimal spanning tree of the graph G. Prove that there is a cut of G, such that (u, v) is a light crossing edge for that cut.

(b) Prove that, if for every cut in G there exists a unique light crossing edge for that cut, then G contains a unique minimal spanning tree. Also prove that the converse is not true in general.

Exercise G3 (Spanning trees III)
Let $e = (i, j)$ be an edge of minimal weight in an undirected, connected, weighted graph $G = (V, E)$. Prove the existence of a minimal spanning tree containing that edge (i, j). Does every minimal spanning tree have to contain that edge?

Homework

Exercise H13 (Finding anagrams) (10 points)
In this exercise we want to work with anagrams. Two words A and B are anagrams of each other, if they contain the same characters with same cardinality of these characters and without respecting the difference of small and capital characters. Examples are the words 'meat'and 'team', 'dog' and 'god' or 'plates' and 'staples'. An example for words not being anagrams could be 'mama' and 'beer'.

Now assume we have a dictionary containing many words and we have to find out which words are anagrams of each other. Construct an algorithm solving this problem. You can describe it pseudocode or just by words. The description should be sufficiently precise, such that a fellow student of yours would be able to implement the algorithm by just using you description. What is the worst case runtime of your algorithm? On which property (parameter) of the set of given words does the algorithms runtime also depend?

Exercise H14 (Trees and forests) (10 points)

(a) Prove that you can use an algorithm finding minimal spanning trees, to find a forest with maximal weight.

(b) Prove that conversely you can use an algorithm finding maximal forests, to find a minimal spanning tree, if the graph is connected.

Exercise H15 (Updates on spanning trees) (10 points)
Let S be a minimal spanning tree of an undirected, connected, weighted graph $G = (V, E)$. Find an algorithm which determines a minimal spanning tree in the new graph G' arising from the following modifications:

(a) Delete an edge $(i, j) \in E$. Assume that the resulting graph $G' = (V, E')$ is still connnected.

(b) Add an edge $(i, j) \in E$. Assume that G is not a complete graph.

Prove the correctness of your algorithms and determine its runtimes.