Homework 4

CS 4104 (Fall 2009)

Assigned on Monday, September 28, 2009.
Hardcopy due at the beginning of class on Monday, October 5, 2009.

Instructions:

- Do not forget to typeset your solutions. *Every mathematical expression must be typeset as a mathematical expression, e.g., the square of n must appear as n^2 and not as "nˆ2".* Students using LaTeX or LyX can use the corresponding versions of the homework problems to start entering their solutions.
- Describe your algorithms as clearly as possible. The style used in the book is fine, as long as your description is not ambiguous. Explain your algorithm in words. A step-wise description is fine. *However, if you submit detailed pseudo-code without an explanation, we will not grade your solutions.*
- Do not make any assumptions not stated in the problem. If you do make any assumptions, state them clearly, and explain why the assumption does not decrease the generality of your solution.
- Do not describe your algorithms only for a specific example you may have worked out.
- You must also provide a clear proof that your solution is correct (or a counter-example, where applicable). Type out all the statements you need to complete your proof. *You must convince us that you can write out the complete proof. You will lose points if you work out some details of the proof in your head but do not type them out in your solution.*
- Analyse your algorithm and state the running time. You will only get partial credit if your analysis is not tight, i.e., if the bound you prove for your algorithm is not the best upper bound possible.

Problem 1 (35 points) Let $G = (V, E)$ be an undirected graph and let $c : E \rightarrow^{+}$ be a function specifying the costs of the edges. Assume that no two edges have the same cost. Given a set $S \subset V$, where S contains at least one element and is not equal to V, let e_S denote the edge in E defined by applying the cut property to S, i.e.,

$$e_S = \arg\min_{e \in cut(S)} c_e.$$

In this definition, the function arg min is just like min but returns the argument (in this case the edge) that achieves the minimum. Let F be set of all such edges, i.e., $F = \{e_S, S \subset V, S \neq \emptyset\}$. Answer the following questions, providing proofs for all but the first question.

(i) (5 points) How many distinct cuts does G have?

(ii) (10 points) Consider the graph induced by the set of edges in F, i.e., the graph formed by $G' = (V, F)$. Is G' connected?

(iii) (10 points) Does G' contain a cycle?

(iv) (5 points) How many edges does F contain?

(v) (5 points) What conclusion can you draw from your answers to the previous statements?

Problem 2 (25 points) Solve exercise 21 in Chapter 4 (page 200) of your textbook. Note that you cannot sort the edges, since your algorithm must run in $O(n)$ time. (*Hint:* Use the solution to exercise 2 in Chapter 3 (page 107) of your textbook. If you like, you can give your algorithm for this exercise some name, e.g., `FindCycle`, and simply use this algorithm as a sub-routine. You do not have to describe the `FindCycle` algorithm.)

Problem 3 (40 points) You are given a graph $G(V, E)$ and a minimum spanning tree (V, T) for G. You now pick an edge e in E and change its cost from c to c'. Describe an algorithm to update T so that it is a minimum spanning tree for the graph with the new edge weight. Your algorithm must run in time linear in the number of nodes and edges in G. There are multiple cases to consider, depending on whether e is in T or not and whether c is larger or smaller than c'. You may find it useful to describe each case separately. You can assume that all edge costs are distinct, both before and after the change.