Homework 2

CS 4104 (Spring 2016)

Assigned on Tuesday, February 9, 2016.
Submit a PDF file containing your solutions on Canvas by the beginning of class on Thursday, February 18, 2016.

Instructions:

- You can pair up with another student to solve the homework. You are allowed to discuss possible algorithms and bounce ideas with your team-mate. **Do not discuss proofs of correctness or running time in detail with your team-mate.** Please form teams yourselves. Of course, you can ask me for help if you cannot find a team-mate. You may choose to work alone. *Each of you must write down your solution individually, and write down the name of the other member in your team. If you do not have a team-mate, please say so.*
- Apart from your team-mate, you are not allowed to consult any sources other than your textbook, the slides on the course web page, your own class notes, the TAs, and the instructor. In particular, do not use a search engine.
- 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 "n^2".* Students can use the LaTeX version 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.*
- Describe an analysis of your algorithm and state and prove 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 (25 points) Solve exercise 2 in Chapter 3 (page 107) of your textbook. Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, it should output one. (It should not output all the cycles in the graph, just one of them.) The running time of your algorithm should be $O(m + n)$ for a graph with n nodes and m edges. Your proof of correctness must consider two cases:

(a) The graph does not contain a cycle: in this case, you should prove that your algorithm correctly reports this fact.

(b) The graph contains one or more cycles: in this case, you must prove that the algorithm correctly computes one of the cycles in the graph. If the graph contains many cycles, it is enough to report one of the cycles.

Problem 2 (25 points) Solve exercise 5 in Chapter 3 (page 108) of your textbook. A binary tree is a rooted tree in which each node has at most two children. Show by induction that in any binary tree, the number of nodes with two children is exactly one less than the number of leaves.

To get started, given a tree T, let us define two useful quantities: c_T (c with T as a subscript) the number of nodes in T with two children, and l_T (l with T as a subscript) the number of leaves in T. With these two quantities defined, the goal of the problem is to prove that for every tree T, $c_T = l_T - 1$. To assist you with the proof, here are three candidates for the induction hypothesis. In your solution, state which candidate is correct and then provide the complete proof by induction based on this choice.

(i) There exists a tree T on $n-1$ nodes such that $c_T = l_T - 1$.

(ii) For every integer k between 1 and $n-1$, there exists a tree T on k nodes such that $c_T = l_T - 1$.

(iii) For every tree T on $n-1$ nodes, $c_T = l_T - 1$.

Problem 3 (20 points) Solve exercise 7 in Chapter 3 (page 108-109) of your textbook. I provide the essential part of the problem here.

> *Claim: Let G be an undirected graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected.*

Decide whether you think the claim is true or false, and give a proof of either the claim or its negation.

Problem 4 (30 points) Solve exercise 9 in Chapter 3 (page 110) of your textbook. Suppose that an n-node undirected graph G contains two nodes s and t such that the distance between s and t is strictly greater than $n/2$. Show that G must contain some node v, not equal to s or t, such that deleting v from the graph destroys all s-t paths. (In other words, the graph obtained from G by deleting v contains no path from s to t.) Give an algorithm with running time $O(m+n)$ to find such a node v. *Hint:* Many of the layers in the BFS tree rooted at s have a special property.