Homework 2

CS 4104 (Fall 2014)

September 23, 2014

Homework is due on 2nd October, 2014

Problem 1 (10 points) Solve exercise 2 in Chapter 3 (page 107) of your textbook. 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 (10 points) Solve exercise 5 in Chapter 3 (page 108) of your textbook. Given a tree $T$, let us define two useful quantities: $c_T$ (with $T$ as a subscript) the number of nodes in $T$ with two children, and $l_T$ (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 (10 points) Solve exercise 7 in Chapter 3 (page 108-109) of your textbook.

Problem 4 (10 points) Solve exercise 9 in Chapter 3 (page 110) of your textbook. Hint: Many of the layers in the BFS tree rooted at $s$ have a special property.

Problem 5 (20 points) Describe an algorithm that uses a priority queue (heap) to merge $k$ sorted lists into one sorted list. Each sorted list has $n$ integers. Neither $k$ nor $n$ is a constant.

To get you started, consider the following algorithm:

1. Insert all the $kn$ numbers into a priority queue, with each number being its own key.

2. Repeatedly report the smallest key in the queue and delete this value from the queue, until the queue becomes empty.

What is the running time of this algorithm?

Now develop an algorithm that has a better running time. Describe your algorithm, prove its correctness, and provide an analysis of the running time. Don’t forget to show that your algorithm has a faster running time than the one described above!