CSCE629 Analysis of Algorithms

Homework 3

Texas A&M U, Fall 2019 *09/13/19*

Lecturer: Fang Song *Due: 10am, 09/20/19*

Instructions.

- Typeset your submission by LaTeX, and submit in PDF format. Your solutions will be graded on *correctness* and *clarity*. You should only submit work that you believe to be correct, and you will get significantly more partial credit if you clearly identify the gap(s) in your solution. You may opt for the "I'll take 15%" option (details in Syllabus).
- You may collaborate with others on this problem set. However, you must **write up your own solutions** and **list your collaborators and any external sources** for each problem. Be ready to explain your solutions orally to a course staff if asked.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

This assignment contains 4 questions, 5 pages for the total of 60 points and 10 bonus points. A random subset of the problems will be graded.

1. (Akinator's trick) Play the game Akinator online (`https://en.akinator.com/`), and answer the questions below.

   (a) (10 points) Given a *sorted* array A with distinct numbers, we want to find out an i such that $A[i] = i$ if exists. Give an $O(\log n)$ algorithm.

   (b) (10 points) Consider a sorted array with distinct numbers. It is then rotated k (k is unknown) positions to the right, and call the resulting array A. (Example: (8,9,2,3,5,7) is the sorted array (2,3,5,7,8,9) rotated to the right by 2 positions) Design as efficient an algorithm as you can to find out if A contains a number x.

   Exercise (do not turn in). Can you think of some real-world problems that the techniques in your algorithms could be useful?

2. (Counting inversions) Given a sequence of n *distinct* numbers $a_1, \ldots, a_n$, we call (a_i, a_j) an *inversion* if $i < j$ but $a_i > a_j$. For instance, the sequence $(2, 4, 1, 3, 5)$ contains three inversions $(2, 1)$, $(4, 1)$ and $(4, 3)$.

   (a) (15 points) Given an algorithm running in time $O(n \log n)$ that counts the number of inversions. (Hint: does Merge-sort help?) Can you also output all inversions?

   (b) (10 points (bonus)) Let's call a pair a *significant inversion* if $i < j$ and $a_i > 2a_j$. Given an $O(n \log n)$ algorithm to count the number of significant inversions.

3. (10 points) (Cycles) Give an algorithm to detect whether a given undirected graph contains a cycle. If the graph contains a cycle, then your algorithm should output one (not all cycles, just one of them). The running time of your algorithm should be $O(m+n)$ for a graph with n nodes and m edges.

4. (15 points) (Shortest cycles containing a given edge) Give an algorithm that takes as input an undirected graph $G = (V, E)$ and an edge $e \in E$, and outputs a shortest cycle that contains e (if no cycle containing e exists, the algorithm should output "none").

   (Note: Give as efficient an algorithm as you can.)