CS180 Homework 6

Due: 4:00pm, 05/11/2016

1. For a directed graph, the *underlying undirected graph* is a graph that is obtained by ignoring the direction of the edges or replacing each directed edge by an undirected edge. We defined a *tournament graph* to be a directed graph that its underlying undirected graph is a complete graph (i.e. if we ignoring the direction there is an edge between every pair of nodes).

   (a) A *king* in a graph is a node that can reach any other node by a path of length ≤ 2. Prove that there is a king in the tournament graph.

   (b) Design an algorithm that finds a king in any tournament graph. Your algorithm should be linear time in the number of edges.

2. Given a real number X and two sets S_1 and S_2 that contain some real numbers.

   (a) Design an $O(n \log n)$ algorithm to find whether two sets contain a same number, where n is the total number of elements.

   (b) Design an $O(n \log n)$ algorithm to find whether there exists a number from S_1, and an number from S_2, whose sum is exactly X. (Hint: reduce the problem to the above problem)

   (c) Given an input set S containing n real numbers, and a real number X.

      i. Design an algorithm to determine whether there are two elements of S whose sum is exactly X. The algorithm should run in time $O(n \log n)$.

      ii. Suppose now that the set S is given in a sorted order. Design an algorithm to solve this problem in time $O(n)$.

3. Given a $n \times n$ binary array. Each row represents a binary number, and so there are n numbers in the array. Assume one operation is looking up a bit in the array and the time is measured by the number of such operations.

   (a) Design an $O(n)$ time algorithm to find a number that is different with all n numbers in the array.

   (b) Prove that $\Omega(n)$ is a lower bound on the number of steps required to solve this problem.

4. Modify radix sort to work for variable-length strings and w.l.o.g assume the numbers are given in base n, and there are n numbers. You can *no longer* assume that all numbers have same digits. Some numbers may be long and some short. It is possible of course to "pad" all numbers by adding "dummy" (0) digits to make them all of the same length. Find an algorithm that *avoids* doing that and achieves a running time linear in the total number of digits.

★ Homework assignments are **STRICTLY** due on the exact time indicated. Please submit a hard copy of your homework solution with your **Name**, **Bruin ID**, **Discussion Number**, clearly indicated on the first page. If your homework consists of multiple pages, please **staple** them together. Email attachments or other electronic delivery methods are not acceptable.

★ We recommend to use LaTeX, LyX or other word processing software for submitting the homework. This is not a requirement but it helps us to grade the homework and give feedback. For grading, we will take into account both the correctness and the clarity. Your answer are supposed to be in a simple and understandable manner. Sloppy answers are expected to receiver fewer points.

★ Unless specified, you should justify your algorithm with proof of correctness and time complexity.