CSC-421 Applied Algorithms and Structures
Winter 2019

Instructor: Iyad Kanj
Office: CDM 832
Phone: (312) 362-5558
Email: ikanj@cs.depaul.edu
Office Hours: Monday & Wednesday 4:00-5:30
Course Website: https://d2l.depaul.edu/

Assignment #1
(Due January 25)

Remarks

- For the questions on this assignment, if needed, you may assume that sorting n numbers can be done in time $O(n \lg n)$ (e.g., using Heap Sort). If you need to sort, you can directly apply such a sorting algorithm (without writing the pseudocode), and claim that it runs in $O(n \lg n)$ time, where n is the number of elments/numbers being sorted.
- When asked to give an algorithm that meets a certain time bound, you need to give the algorithm (pseudocode/description) and analyze its running time to show that it meets the required bound; giving only the algorithm is not enough to receive full credit.
- Please upload your submission as a single PDF file on D2L. If your submission consists of more than one file, convert all your files into a single PDF file and upload it.

1. Given a collection of n nuts and a collection of n bolts, arranged in an increasing order of size, give an $O(n)$ time algorithm to check if there is a nut and a bolt that have the same size. The sizes of the nuts and bolts are stored in the sorted arrays $NUTS[1..n]$ and $BOLTS[1..n]$, respectively. Your algorithm can stop as soon as it finds a single match (i.e, you do not need to report all matches).

2. Let $A[1..n]$ be an array of distinct positive integers, and let t be a positive integer.

   (a) Assuming that A is sorted, show that in $O(n)$ time it can be decided if A contains two distinct elements x and y such that $x + y = t$.

   (b) Use part (a) to show that the following problem, referred to as the 3-SUM problem, can be solved in $O(n^2)$ time:

   > 3-SUM
   > Given an array $A[1..n]$ of distinct positive integers that is not (necessarily) sorted, and a positive integer t, determine whether or not there are three distinct elements x, y, z in A such that $x + y + z = t$.

3. Let $A[1..n]$ be an array of positive integers (A is not sorted). Pinocchio claims that there exists an $O(n)$-time algorithm that decides if there are two integers in A whose sum is 1000. Is Pinocchio right, or will his nose grow? If you say Pinocchio is right, explain how it can be done in $O(n)$ time; otherwise, argue why it is impossible.

4. Let $A[1..n]$ be an array of points in the plane, where $A[i]$ contains the coordinates (x_i, y_i) of a point p_i, for $i = 1, \ldots, n$. Give an $O(n \lg n)$ time algorithm that determines whether any two points in A are identical (that is, have the same x and y coordinates).

5. Textbook, page 1066, exercise 34.2-6.

6. Textbook, page 1086, exercise 34.4-5 (look for the definition of disjunctive normal form in chapter 34 of the textbook).

7. Textbook, page 1086, exercise 34.4-6.