Honors Algorithms CMSC-27230 First Quiz. February 6, 2020

Instructor: László Babai

NAME (print with LARGE letters) ______________________________

Are you using a cheat sheet? Circle the answer: YES NO
If YES, do not forget to **hand it in** at the end of the test.

Please DO NOT SIT next to anyone (leave an empy seat). Please do not sit near a person to whom you usually sit near in class.

SHOW ALL YOUR WORK. Do NOT use book, notes, or scrap paper. You may use a "cheat sheet": one page of HANDWRITTEN notes in English, written in ink in your handwriting with your name printed in large English letters on the top. NO PHOTOCOPIES! Hand in your cheat sheet with the test.

The use of ELECTRONIC DEVICES is STRICTLY FORBIDDEN.

Write your answers IN THE SPACE PROVIDED. You may **continue on the reverse.** DO NOT USE paper other than the problem sheet provided.

When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English).

This quiz contributes 6% to your course grade.

1. (18+9+8B points) Let $A[1 \dots n]$ be an array of zeros and ones, with $A[1] = 0$ and $A[n] = 1$. Find an address i $(1 \leq i \leq n-1)$ such that $A[i] = 0$ and $A[i+1] = 1$. In our model, the only way we can access the input is through YES/NO *queries* of this type: $A[j] \stackrel{?}{=} 0$. Design an algorithm that requires the minimum number of queries.

   (a) Describe your algorithm in elegant pseudocode.

   (b) The *query complexity* of your algorithm is a function $f(n)$ defined as the maximum number of queries made by the algorithm, where the maximum is taken over the set of all arrays A of length n. (The number of queries for the "worst" input.) Your answer should be a simple function of n. State, do not prove.

   (c) (Bonus) Prove that your algorithm is optimal. What you need to prove is that *every* algorithm requires at least $f(n)$ queries on some input array.

2. (8+25+8B points) A Divide-and-Conquer algorithm reduces and instance of size n to four instances of size $n/3$. The cost of the reduction is $O(n)$. Let $f(n)$ denote the worst-case complexity of this algorithm, i.e., the maximum cost over all inputs of size n.

   (a) State a recurrent inequality for $f(n)$.

   (b) Prove: $f(n) = O(n^r)$ for some contant r. (r is a real number.) Find the smallest value of r for which this bound follows from the given information. (You don't need to prove that it is the smallest.) State the exact value of r, not an approximation of it. For the proof, assume $n = 3^k$ so no rounding will occur.

   (c) (Bonus) Prove that your value r is the smallest value inferable from the given information.