CS 330, Fall 2019, Homework 2, Due Wednesday, September 18, 2019

Homework Guidelines

Please make sure you read the collaboration policy before you start working on your homework. Refer to the general information handout for the homework policy and additional options.

Collaboration policy Collaboration on homework problems, with the exception of programming assignments and reading quizzes, is permitted, but not encouraged. If you choose to collaborate on some problems, you are allowed to discuss each problem with at most 5 other students currently enrolled in the class. Before working with others on a problem, you should think about it yourself for at least 45 minutes. Finding answers to problems on the Web or from other outside sources (these include anyone not enrolled in the class) is strictly forbidden.

You must write up each problem solution by yourself without assistance, even if you collaborate with others to solve the problem. You must also identify your collaborators. If you did not work with anyone, you should write "Collaborators: none." It is a violation of this policy to submit a problem solution that you cannot orally explain to an instructor or TF.

Solution guidelines For problems that require you to provide an algorithm, you must give the following:

1. a precise description of the algorithm in English and, if helpful, pseudocode,
2. a proof of correctness,
3. an analysis of running time and space.

You may use algorithms from class as subroutines. You may also use any facts that we proved in class.

You should be as clear and concise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an important skill. A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error and because it is easier to read and understand. Points might be subtracted for illegible handwriting and for solutions that are too long. Incorrect solutions will get from 0 to 30% of the grade, depending on how far they are from a working solution. Correct solutions with possibly minor flaws will get 70 to 100%, depending on the flaws and clarity of the write up.

Problems to be handed in

1. (**Asymptotic Analysis**, 10 points) For each of the following statements, decide whether it is always true, never true, or sometimes true for nonnegative functions f and g. If it is always true or never true, give a proof (using the definitions of $O()$, $\Omega()$, $o()$, etc). If it is sometimes true, give one example for which it is true, and one for which it is false.

   (a) $f(n) = o(g(n))$ and $f(n) = \Omega(g(n))$

(b) If $f(n) = O(g(n))$, then $2^{f(n)} = O\left(2^{g(n)}\right)$

(c) If $f(n) = O(g(n))$, then $g(n) - f(n) = \Omega(g(n))$.

(d) If $f(n) = o(g(n))$, then $g(n) - f(n) = \Omega(g(n))$.

2. (**Smallest items**, 10 points)

Give an algorithm that takes a pointer/reference to a min-heap of size n as input, and returns the k smallest items from the heap in time $O(k \log k)$.

(Notice that this running time doesn't depend on n. You can assume that the heap is already in memory, so you don't need to read the whole thing.)

Hint: Explore the heap's tree from the root, in order of key size. How can you decide which node to explore next? You might need an extra data structure to help you out.