CS7800: Advanced Algorithms. Fall 2017
Homework 1

Instructor: Jonathan Ullman   TA: Albert Cheu

Due Friday, September 15 at 11:59pm
(Email to neu.cs7800@gmail.com)

NB: Typically the weekly homework assignments for this class will consist of ≈ 2 problems. This homework is a bit longer because the first two problems are short review questions. A more typical workload would be to do problems 3 and 4. Problems 1-3 review concepts from undergraduate discrete math and algorithms. I recommend starting these problems right away to make sure that you have the appropriate background for this course.

Homework Guidelines

Collaboration Policy. Collaboration on homework problems is permitted, however it will serve you well on the exams if you solve the problems by yourself. If you choose to collaborate, you may discuss the homework with at most 2 other students currently enrolled in the class. 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, indicate that on your submission. If asked, you must be able to explain your solution to the instructors.

Preparing and Submitting Solutions. You must type your solutions using LaTeX. Please use an 11-pt or larger font. Please submit both the source and PDF files using the naming conventions `lastname_hw1.tex` and `lastname_hw1.pdf`. Your name must be on the first page of the PDF.

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. You may use any facts from class in your analysis and you may use any algorithms from class as subroutines in your solution.

You should be as clear and concise as possible in your write-up of solutions. Communication of technical material is an important skill, so clarity is as important as correctness. 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 solutions that are too long.

Review Problems

Problem 1 (Review of Asymptotic Growth, 5 pts)**.** Arrange the following list of functions in *ascending* order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$. (You do not need to provide proofs.)

$$f_1(n) = n! \quad f_2(n) = n^{\log_2(3)} \quad f_3(n) = 20! \quad f_4(n) = \log_2(n^3)$$
$$f_5(n) = 2^{16n} \quad f_6(n) = \sqrt{n}\log_2^2(n) \quad f_7(n) = 4096n \quad f_8(n) = n^{3/2} + n\log_2^2(n)$$

Problem 2 (Review of Basic Proof Techniques, 10 pts)**.**

(a) Prove the following statement by contradiction: Let G be a simple graph on $2n$ nodes with no self-loops[1] Prove by contradiction that if every node in G has degree at least n, then G is a connected graph.

(b) Prove the following statement by induction: Given an unlimited supply of 5 and 8 cent coins, one can make exact change for any amount greater or equal to 28 cents.

Problem 3 (Review of Basic Algorithms, 10 pts)**.** You need to find an apartment. Since you like to be known for finding good deals, it's crucial to you that you find an apartment that is cheaper than each of your neighbors on your street. Unfortunately, the Boston housing market is out of control and most apartments go for well above the listed rent, so you can't know the price of an apartment until you visit it and scout out the other interested renters. Since you are busy with algorithms homework, you don't have time to visit every apartment. Fortunately, we will show that you only need to visit a small fraction of the apartments to find a good deal.

More formally, there are n apartments with prices $p_1, \ldots, p_n$. For simplicity, we will assume that all prices are distinct. A *locally cheapest apartment among* $p_1, \ldots, p_n$ is an apartment i such that $p_i \le p_{i+1}$ and $p_i \le p_{i-1}$. If $i = 1$ then we only require $p_1 \le p_2$ and if $i = n$ then we only require $p_n \le p_{n+1}$. We will represent the process of going to an apartment and finding out the rent as a function $\textsc{RevealPrice}(i)$ that takes as input $1 \le i \le n$ and returns p_i.

Design an algorithm that takes as input an array of numbers representing house prices $p_1, \ldots, p_n$, runs for $O(\log n)$ time, and finds a locally cheapest house. Clearly describe your algorithm, prove that it is correct, and analyze its running time.

Stable Matching

Problem 4 (Preferences with Many Stable Matchings, 5 pts)**.** The Gale-Shapley algorithm demonstrates that at least one stable matching always exists, but there can sometimes be many, even exponentially many, stable matchings. First, give an example of preferences for two men and two women where two distinct stable matchings exist. Then, show that for every n, there exist preferences for n men and n women where there are $\Omega(c^n)$ distinct stable matchings, for some constant $c > 1$. In other words, there can be exponentially many stable matchings. (For bonus points, try to get as large a value of c as you can!)

---

[1] That is, there are no edges connecting a node to itself and there is at most one edge connecting any pair of nodes.