

Homework 1 – Due Friday, September 5, 2008

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.
- Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. *Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.*
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class.

Exercises These should not be handed in, but the material they cover may appear on exams: problems in Chapters 1 and 2.

Problems to be handed in

1. **(Stable Matching with Indifferences)** Chapter 1, problem 5.
If you give an algorithm for either part of the question, please give *both* an English description *and* pseudocode for your algorithm. Analyze its time and space complexity.
2. **(Truthfulness in Stable Matching)** Chapter 1, problem 8. *Hint:* Try playing with several specific examples of preference lists.
3. **(Order of Growth Rate)** Chapter 2, problems 3 and 4. Please add $g_8(n) = n!$ to the list of functions in #4.
4. **(Understanding big-O notation)** Chapter 2, problem 5.
5. * **(Optional: How Many Stable Matchings?)** The analysis of the Gale-Shapley algorithm establishes that every instance of the stable marriage problem admits at least one stable matching. Here we consider *how many* such matchings might exist.
(NB: You must solve both parts of the problem to receive credit.)

- (a) Give an algorithm that takes an instance of the stable marriage problem as input and decides if there is *exactly one* stable matching for this instance (that is, the algorithm outputs either “unique stable matching”, or “more than one stable matching”). Pay close attention to the proof of correctness of your algorithm.
- (b) Show that the maximum number of possible stable matchings for instances with n men and n women scales at least exponentially with n : that is, show that there is a constant $c > 1$, and a sequence of instances of the stable marriage problem, x_1, x_2, \dots , one for each value of n , such that the number of stable matchings in instance x_n is at least c^n . (Extra extra points for a construction with $c > 2$.)