CS 310 — Algorithms — Spring 2014
Programming Assignment #1

Due on LMS by: 25 October 2020 @11pm

1 Description

In this assignment, you will implement a variation of the stable matching problem. We will assume that there are more students graduating than there are slots available in the m hospitals. We want to find a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital)

We say that an assignment of students to hospitals is *stable* if neither of the following situations arises:

- First type of instability: There are students s and s', and a hospital h, such that
  - s is assigned to h, and
  - s' is assigned to no hospital, and
  - h prefers s' to s
- Second type of instability: There are students s and s', and hospitals h and h', so that
  - s is assigned to h, and
  - s' is assigned to h', and
  - h prefers s' to s, and
  - s' prefers h to h'.

So we basically have the Stable Matching Problem as presented in class, except that (i) hospitals generally want more than one resident, and (ii) there is a surplus of medical students. There are two parts of this problem.

1. Write a short report that includes the following information:
   (a) Show that there is always a stable assignment of students to hospitals.
   (b) Give an algorithm in pseudocode (either an outline or paragraph works) to find a stable assignment. *hint: it should be very similar to the Gale-Shapley Algorithm*
   (c) Give a proof of your algorithm's correctness. Remember that you must prove both that your algorithm terminates and gives a correct result.
   (d) Give the runtime complexity of your algorithm in Big O notation and explain why.
   (e) Consider a Brute Force Implementation of the algorithm where you find all combinations of possible matchings and verify if they are a stable marriage one by one. Give the runtime complexity of this brute force algorithm in Big O notation and explain why.
   (f) In the following two section you will implement a solution. In your report, use the 8 provided data files to plot the number of hospitals (x-axis) against the time in ms it takes for your code to run (y-axis).
2. Implement the efficient algorithm you devised in your report. You can use any language. The end result should be a program that can be given one of the provided input files on standard input and it generates the data in the corresponding output file on standard output. Please see input and output file format below.

2 Input file format

First line of input has two integers m and n separated by space where m is the number of hospitals and n is the number of residents.

Second line of input has m integers where the k^{th} integer is the number of slots in the k^{th} hospital.

Next m lines are hospital preferences each containing n integers where the first integer is the index of the most preferred resident and the last integer is the index of the least preferred resident.

Last n lines are resident preferences each containing m integers where the first integer is the index of the most preferred hospital and the last integer is the index of the least preferred hospital.

3 Output file format

Output n integers where the i^{th} integer is the hospital assigned to the i^{th} resident. Use -1 for a resident not assigned to any hospital.