

Design and Analysis of Algorithms

CSCI 3650

Spring 2021

Assignment 1

Implementation of Gale-Shapley Algorithm for Stable Matching Problem

Type: Programming

Due Date: Wed. 10th Feb.

Objective: The objective of this assignment is to give you practice in implementing a fairly simple algorithm using appropriate data structures.

Your Task: For this assignment you will need to implement Gale-Shapley Algorithm for finding a stable matching. For simplicity, instead of using names, we will assume that the men are numbered 1 through n and similarly, the women are numbered 1 through n . Your program should output the man-optimal stable matching produced by the Gale-Shapley Algorithm. Your implementation should be running in $O(n^2)$ time.

You are expected to write the whole program (including implementation of data structures involved) yourself and not borrow any existing classes or functions from the libraries or from public domain out there.

Data Structures: I want to make sure that all of you are comfortable in dealing with different data structures. Use a linked list to represent the preference lists of men. Use an array data structure to represent the preferences of women. Use a stack data structure (that is implemented either as an array or as a linked list) to hold the set of unmatched men.

Input Format: You will be given the preference lists of men and women as two separate files. The second command line parameter will specify the name of the file in which men's preferences can be found. The third command line parameter will specify the name of the file in which women's preferences can be found. Do not hard code names of files in your program. A sample input file showing preferences of men for the case of $n = 4$ is shown below. In the very first line, the number of men is shown. The successive lines following it represent the preference of Man 1, Man 2 etc (the preferences are separated by a space). In the following example, Man 1's first preference is Women 2, second preference is Women 3 and so on.

```
4
2 3 1 4
4 1 2 3
3 1 4 2
2 4 1 3
```

The format for the women's preference file is identical. You can assume that the data files are consistent and don't have to check for their integrity.

Output Format: Your program should create a file whose name is specified as the fourth command line parameter and place the output in that file. Do not hard code file names in your program. The output should simply be n man-woman pairs, one per line. A sample output is shown below.

```
1 4
2 2
3 1
4 3
```

Sample Command Lines: Suppose your executable is called "stable", then

```
stable boys girls matching
```

should get the men's preferences from the file "boys" and should get the women's preferences from the file "girls" and should place the stable matching found in the file "matching".

General remarks: Make sure that your programs are written in C, C++ or Java and that they can run on the `xlogin` system (that runs SUSE Linux OS) of the department of computer science. Make sure to test your program on several test cases so that you can reasonably be confident that it is correct. While implementing, it is very easy to code certain segments of the algorithm which are not spelt in full detail in such a way that the complexity of your program becomes more than the specified complexity of the algorithms. Be extra careful to avoid such pitfalls.

Create a brief "readme" file which has instructions on how one should compile and execute your program, any assumptions that you made beyond what is explicitly given in the handout and any limitations your program has beyond what is allowed as per this handout. Make sure that your program has traditional comments such as name, date, purpose etc at the beginning. Also, comment your program fairly liberally so that a reader can understand the logic of your program easily. Make sure that your programs are indented consistently so that it is easy on the eyes to view it and also easy to follow the logic.

What to submit? Do not submit hard copies of your program. Do not submit your programs on CANVAS. Handin the files you want to submit by using the *submit* utility. To submit your files use the following command

```
~gopal-assignments/csci3650-sp21-mycolor/bin/submit 1 stable.java readme
```

Note that `mycolor` in the above command is a placeholder and should be replaced by either `purple` or `gold` depending on the group to which you belong.

The first component invokes the submit utility. The second component is the assignment number. The last component is the list of files (separated by blanks) that you wish to submit.