

Name: _____

100 points total

CS 2123 Programming Project 1 Fall 2019

Assignment is due at 11:59pm on September 12. Submit a digital copy of the assignment on Harvey. You may submit a lateness coupon request BEFORE the assignment is due by sending an email to cs2123f19@googlegroups.com with Subject "CS2123 Project Lateness Coupon". All other late work will receive a 10 percentage point deduction per day (including weekends), No late work is accepted beyond five days after the assignment is due.

The Gale-Shapley algorithm can be modified to accommodate many variations on the Stable Matching Problem. Your task is to implement the following two variations in Python. Starter code is available at http://secon.utulsa.edu/cs2123/code/gs_starter.py. Attach a printout of your code as well as output running your code from the command line (i.e., so that the code inside the `__main__` portion is executed).

You should turn in a copy of the code and output in a .zip file on Harvey. The Python file must be named `gs.py`, and you must not modify the names of any of the functions or their defined parameters. Any outputs from running the code should be included in a file called `gs_output.txt`, also included in the zip file. **Both of these files must be included in a directory called LastnameFirstname.** Please write your name(s) as a comment in the first line of code in `gs.py` and at the top of the output file.

Note: you are permitted to reuse code from the `gs` method as appropriate.

- a. **Forbidden matches:** In this variant of the Stable Matching Problem, in addition to the set of men M and women W , there is a set $F \subseteq M \times W$ of forbidden pairs that are not allowed to be matched together. Using the revised definition of stable matching on p. 20 of *Algorithm Design*, implement `gs_block(men, women, pref, blocked)`, where `blocked` is a set of forbidden pairs, for example, of the form:

```
blocked = {('xavier', 'clare'), ('zeus', 'clare'), ('zeus', 'amy')}
```

- b. **Coping with indifference:** In this variant of the Stable Matching Problem, ties are allowed in the preference ordering. They are implemented by turning the preference list into a list of sets, where ties are placed into the same set. Implement `gs_tie(men, women, preftie)`, where `preftie` is a dictionary mapping people to a list of sets, for example, of the form:

```
thepreftie = {'xavier': [{ 'bertha' }, { 'amy' }, { 'clare' }],
              'yancey': [{ 'amy', 'bertha' }, { 'clare' }],
              'zeus':  [{ 'amy' }, { 'bertha', 'clare' }],
              'amy':  [{ 'zeus', 'xavier', 'yancey' }],
              'bertha': [{ 'zeus' }, { 'xavier' }, { 'yancey' }],
              'clare': [{ 'xavier', 'yancey' }, { 'zeus' }]}
}
```

Here, Xavier strictly prefers Bertha to Amy to Clare, Yancey is indifferent between Amy and Bertha, but prefers both to Clare, and so on.