Project Description

We'll implement the Stable Matching algorithm from the previous lesson.

Recall the pseudocode of the algorithm:

While there exists an unmarried man:

1. Pick an arbitrary unmarried man M

2. Choose the top woman W from his list to whom he hasn't proposed yet

3. If W is free or prefers M over her current husband, then marry M and W

We'll write a Python function stableMatching(n, menPreferences, womenPreferences) that gets the number n of women and men, preferences of all women and men, and outputs a stable matching.

For simplicity we'll be assuming that the names of n men and n women are 0, 1, ..., n-1.

Then the menPreferences is a two-dimensional array (a list of lists in Python) of dimensions n by n, where menPreferences[i] contains the list of all women sorted according to their rankings by the man number i. As an example, the man number i likes the best the woman number menPreferences[i][0], and likes the least the woman number menPreferences[i][n-1]. Similarly, the array womenPreferences contains rankings of men by women. For example, womenPreferences[i][0] is the number of man who is the top choice for woman i.

Our function will return a list of length n, where ith element is the number of woman chosen for the man number i.

For convenience we can store

1. unmarriedMen -- the list of currently unmarried men;

2. manSpouse -- the list of current spouses of all man;

3. womanSpouse -- the list of current spouses of all woman;

4. nextManChoice -- contains the number of proposals each man has made.