**SUBJECT**: DESIGN AND ANALYSIS OF ALGORITHMS

**CODE**: 503040

Duration: 150 minutes

Allowed to use materials.

**LAB 09: Iterative Improvement**

# Objectives

Understand the properties of iterative improvement paradigm for algorithm design

Be able to design, implement, and analyze iterative improvement algorithms solving common problems.

# Idea of Iterative Improvement

1. Start with some solution
2. Iteratively improve the solution

# Stable Marriage Problem

## Definition

There are n men and n women.

Each man has an order of preference among the women (no ties). M(i,j) denotes the preference order for i-th man for the j-woman.

Each woman has an order of preference among the men (no ties). W (i, j) denotes the preference order for i-th woman for the j-man.

Thus, for all i,j, M(i,j),W(i,j) ∈ {1,2,...,n} and for all i, for all j,j’ such that j≠ j’, M(i,j)≠ M(i,j’) and W(i,j)≠ W(i,j’).

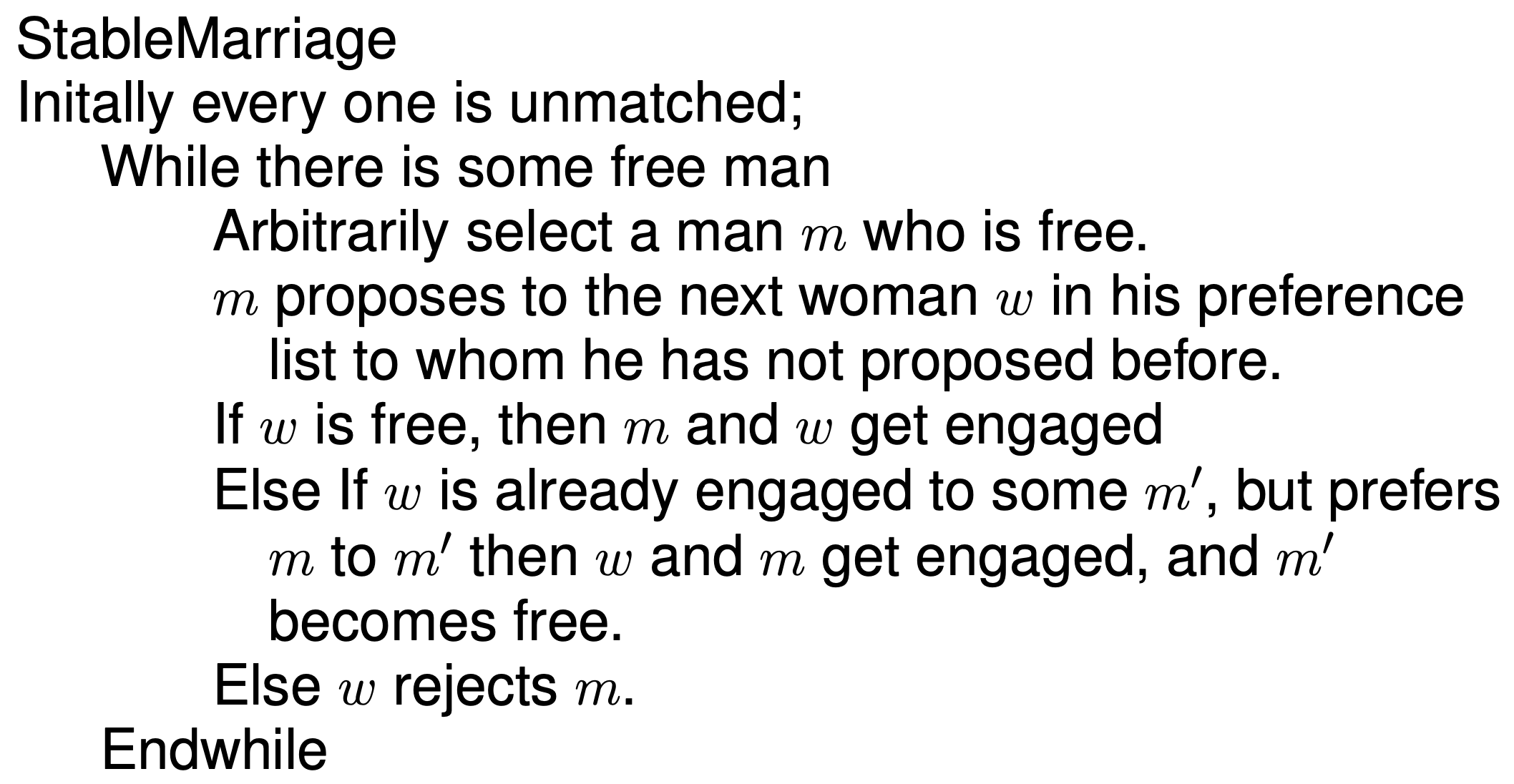
Find a stable-matching: i.e., a bijective function

S : {1,2,...,n} to {1,2,...,n} such that

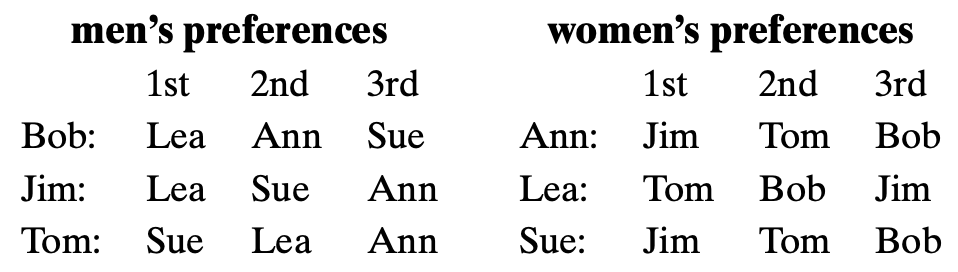
If S(i) = j and S(i’) = j’, then ¬[M(i’,j) < M(i’,j’) and W(j,i’) < W(j,i)]

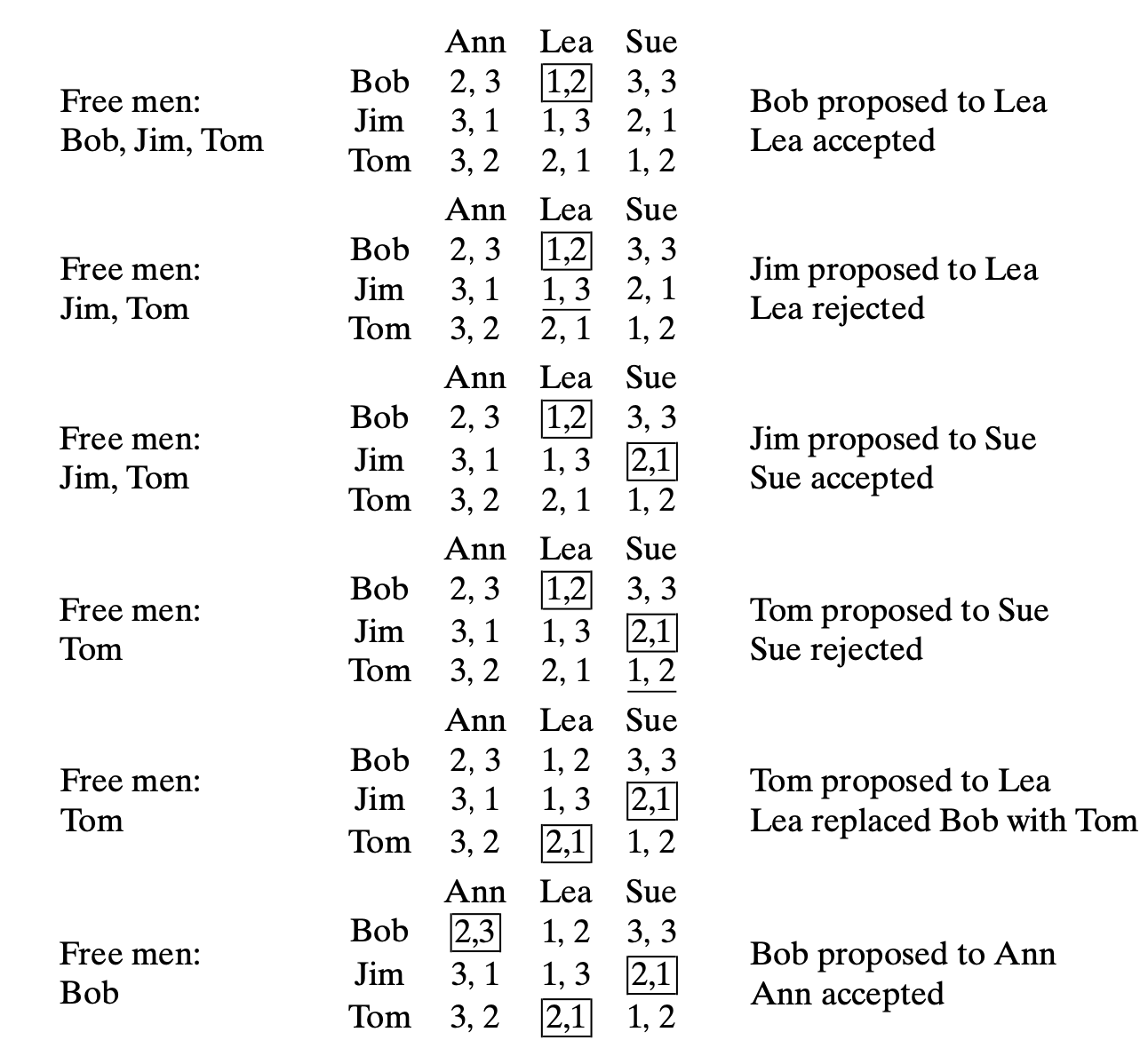
That is, if one matches i-th man to j-th woman and i’-th man to j’-woman, then it is not the case that i’-th man prefers j-th woman more than j’-th woman and j-th woman prefers the i’-th man more than i-th man: in which case there is motivation for both i’-th man and j-th woman to switch.

## An iterative improvement algorithm to solve the problem



## Example of applying the iterative improvement algorithm to the problem





# Exercise

For the problem “Stable Marriage”, implement the presented iterative improvement algorithm to solve the problem.

1. Create Python classes **Man, Woman** to store following information of each person:
2. **name** (str)
3. **preference** (Dict[str, int]): dictionary of entries. Each entry has key=name, value=preference\_order (e.g. 1,2,3)
4. **status** (Option[str]): if unmatched, then status=None, otherwise, the name of the partner of the opposite gender.
5. **unproposed\_preference** [for man only] (List of str): list of women to whom he has not proposed before.
6. Create two lists of unmatched men and women, respectively, in the above example.
7. Write a Python function select to arbitrarily select a man who is free from an input list of men.
8. Write a Python function propose to model proposing process. The function takes lists of men and women and a currently considered man. It returns the changed lists of men and women with their corresponding changed status.
9. Write a complete program to solve the Stable Marriage problem.
10. Test the program on the example.