Stable Matching Problem

Problem (Stable Matching (informal))

Given *n companies and m coop students,*

- *each company ranks the students based on their application (assume each student applied to each company), and*
- *each student ranks the companies based on his/her interests.*

Find an assignment of students to companies that is **perfect***: "all positions filled and all students assigned" and* **stable***: "no pair of a company and a student have a reason to make a new deal."*

Definition (Instability)

All of the following holds:

- Student s_1 is assigned to company c_1.
- Student s_2 is assigned to company c_2.
- s_2 prefers c_1 over c_2
- c_1 prefers s_2 over s_1

Assumption: Assume each company has exactly one coop position (removing asymmetry).

Exercise

What condition n and m must satisfy if we want a *perfect* assignment?

We have a set M of n men (companies) and a set W of n women (students).
Every man **ranks** every woman, and vice versa.
The goal is to pair the men and women ("getting them married").

Definition (Matching)

A **matching** S is a set of ordered pairs from $M \times W$ such that each man and each woman appears at most once in S ("a collection of marriages between the men and the women").

Definition (Perfect matching)

A matching S is **perfect** if each man and each woman appear in **exactly one** pair in S.

Definition (Stable matching)

A matching S is **stable** if it is **perfect** and there is no **instability**.

Exercise

Rewrite the definition of instability using M, W, S notation.

Problem (Stable Marriage)

Given n men and n woman with their preferences, find a matching S that is **stable**.

Exercise

What is the solution if we only need a **perfect matching**?

Exercise

1. Generate an instance of this problem with 4 men m_1, m_2, m_3, m_4 and 4 women w_1, w_2, w_3, w_4. For each man and woman, specify their preferences.
2. Find a solution (stable matching) for your instance.
3. Think about an algorithm that could solve any instance to this problem.

Definition (Algorithm)

a finite set of instruction such that:

1. each step is stated precisely,
2. the result of each step is uniquely defined and depends only on the input and the results of the previous steps,
3. it stops after finitely many steps on every instance of the problem.