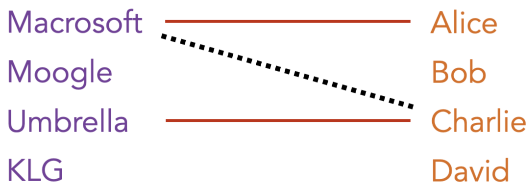
Stable Matching Problem

2-sided markets:

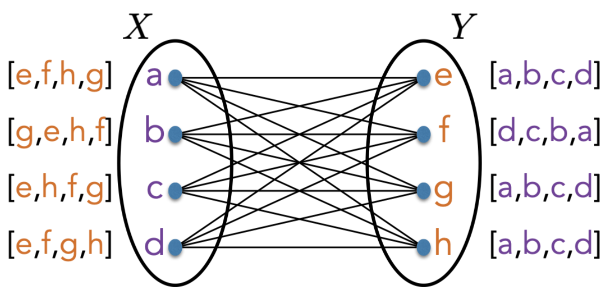
Examples:

Goal:

The problem we want to avoid:



Formalizing the problem. An instance of the problem can be represented as a complete bipartite graph, plus, a preference list for each node.



We assume $|X| = |Y|$.

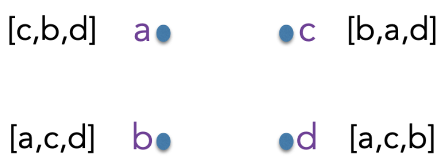
Goal is to find a **stable matching.**

The properties of a stable matching:

1.

2.

A minor digression: What if the graph is not bipartite?



Does this have a stable matching?

Trivial algorithm for solving the stable matching problem:

Gale-Shapley Algorithm.
While there is a company x that is not matched:

-
-

Theorem.

Proof. We need to show 3 things:

1.
2.
3.

We prove each part separately.

<u>Proof of Part 1:</u>

<u>Proof of Part 2:</u> This is a proof by contradiction, so assume we don't have a perfect matching. This means there must exist a company x^* that is not matched. Our strategy to reach a contradiction is to show:

A company x^* is not matched $\implies$ All students must be matched $\implies$ All companies must be matched

- Second implication:

- First implication:
  Observation:

  A company x^* got rejected by every student. Two possible reasons to get rejected:

<u>Proof of Part 3:</u>
Improvement Lemma:

(i)

(ii)

Consider any unmatched pair (x, y). We want to show it cannot be unstable.

- Case 1:

- Case 2:

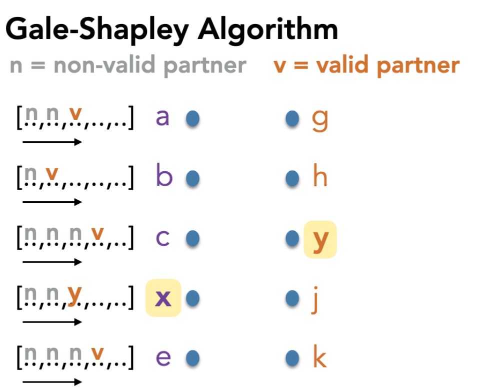
Some interesting questions.
Does the order of how we pick companies matter?
Would it lead to different matchings?
Is the algorithm "fair"? Does it favor companies or students?

Definition (valid partner).

Definition (best(x)).

Theorem.

Proof.



Consider the **first time** an x gets rejected by a valid partner y.

Definition (worst(y)).

Theorem.

Proof. (exercise)