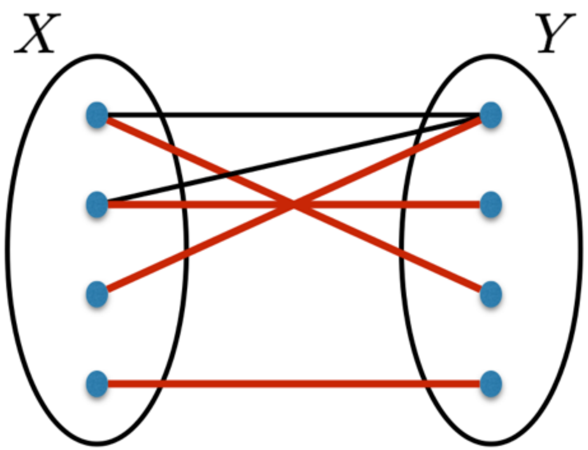
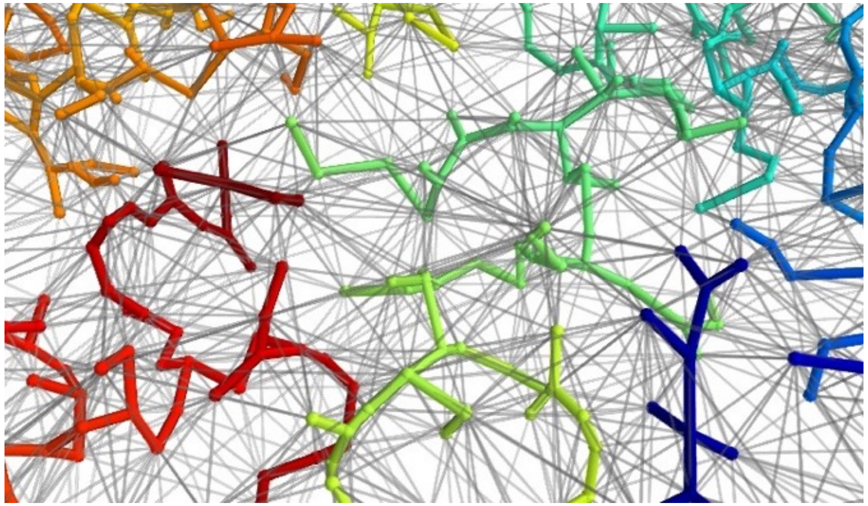
15-251
Great Theoretical Ideas in Computer Science

Lecture 13:
Graphs III: Maximum Matchings





February 28th, 2017

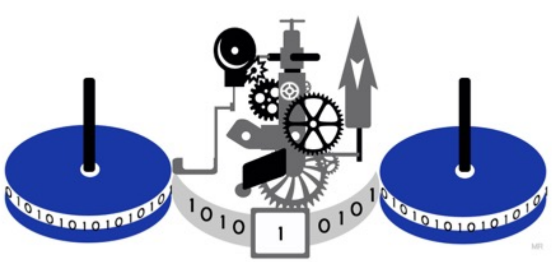
Some motivating real-world examples

matching machines and jobs

Job 1

Job 2

⋮ ⋮



Job n

Some motivating real-world examples

matching professors and courses

⋮

15-110

15-112

15-122

15-150

15-251

⋮

Some motivating real-world examples

matching rooms and courses

GHC 4401	15-110
DH 2210	15-112
GHC 5222	15-122
WEH 7500	15-150
DH 2315	15-251
⋮	⋮

Some motivating real-world examples

matching students and internships

Macrosoft

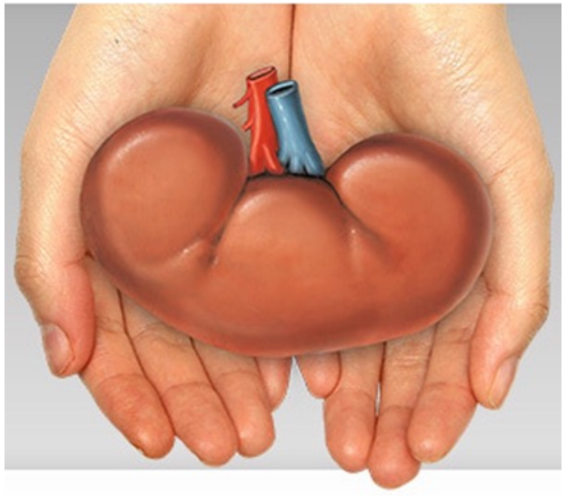
Moogle Google

UMBRELLA CORPORATION

"Our business is life itself..."

Some motivating real-world examples

matching kidney donors and patients



How do you solve a problem like this?

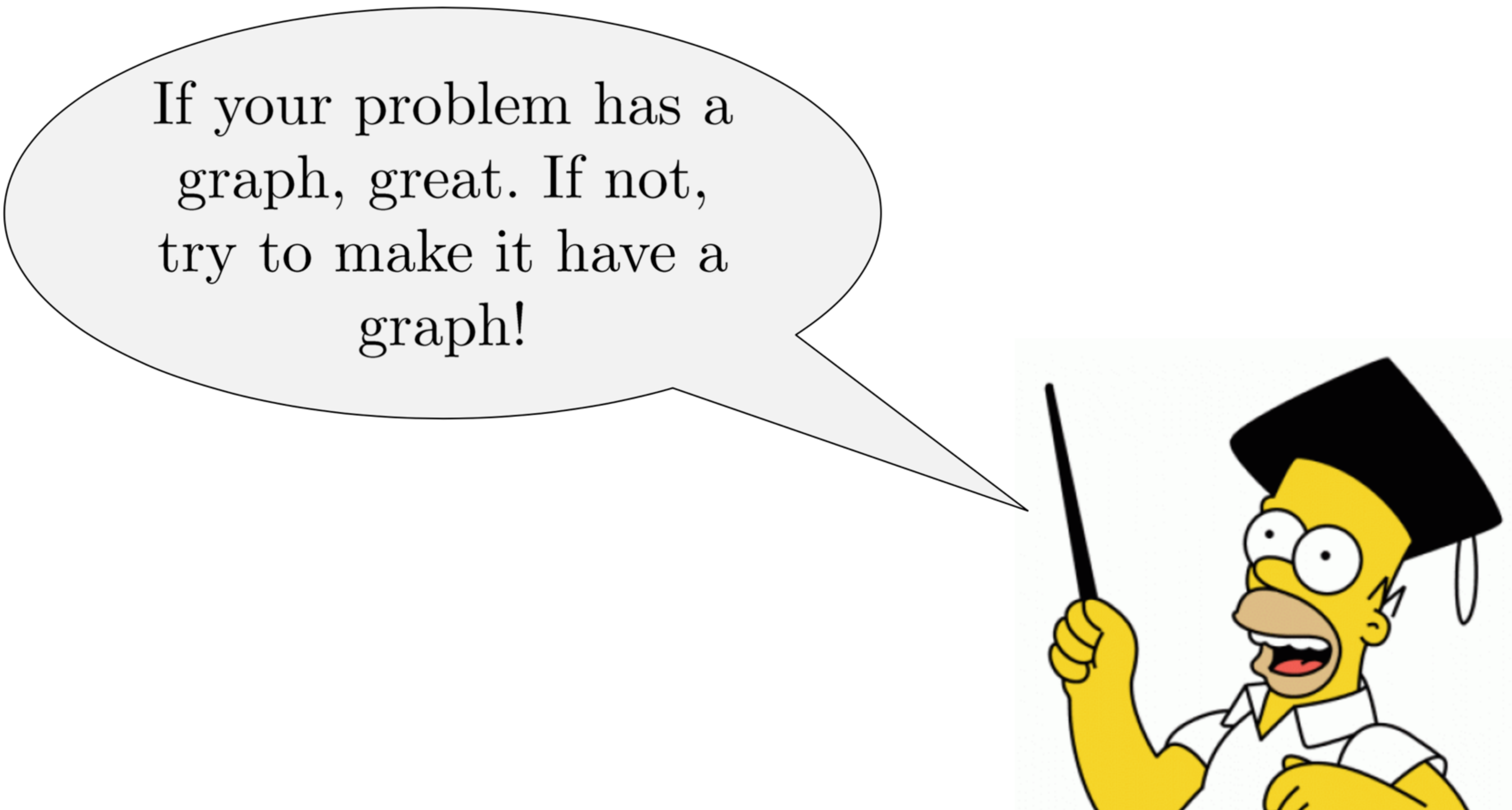
1. Formulate the problem

2. **Ask**: Is there a trivial algorithm?

3. **Ask**: Is there a better algorithm?

4. Find and analyze

Remember the CS life lesson

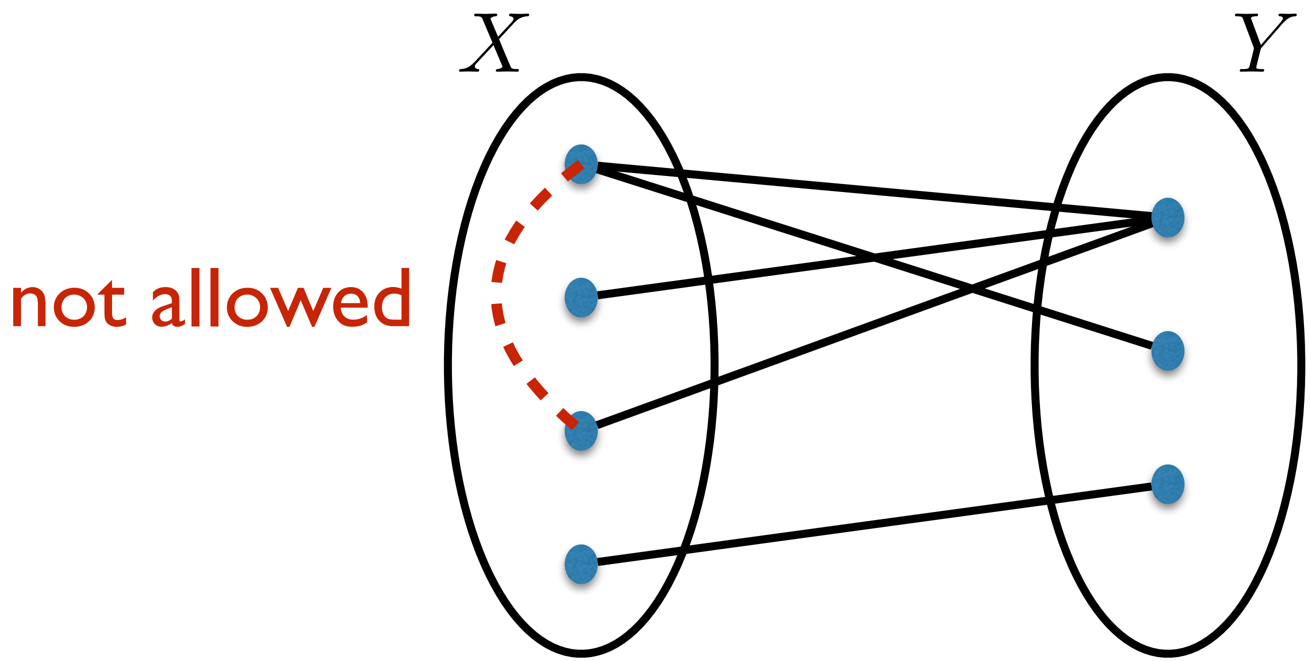


First step: Formulate the problem

Purpose:

- Get rid of all the distractions
- Identify the crux of the problem
- Get a clean mathematical model that is easy to reason about.

Bipartite Graphs



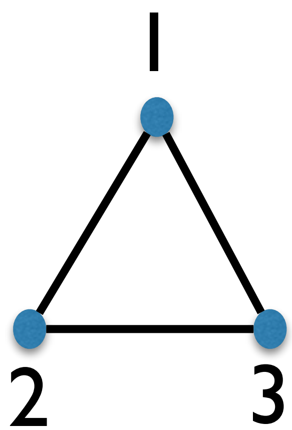
$G = (V, E)$ is **bipartite** if:

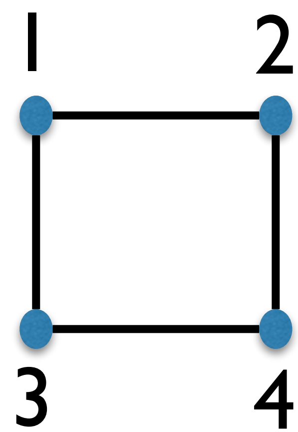
- there exists a bipartition of V into X and Y
- each edge connects a vertex in X to a vertex in Y

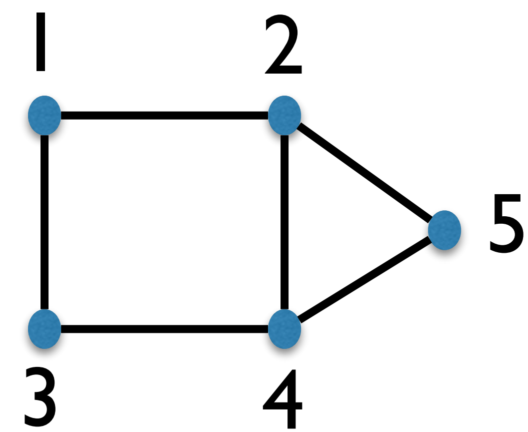
Given a graph $G = (V, E)$, we could ask, is it bipartite?

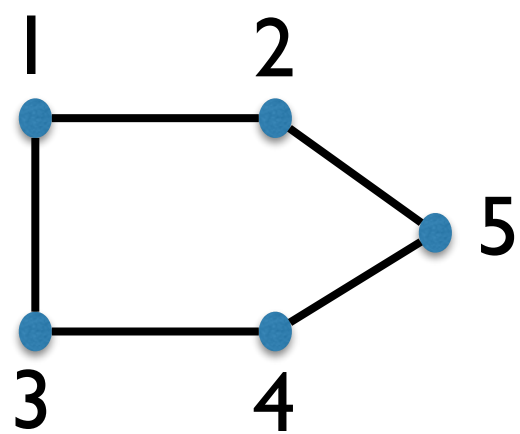
Bipartite Graphs

Given a graph $G = (V, E)$, we could ask, is it bipartite?



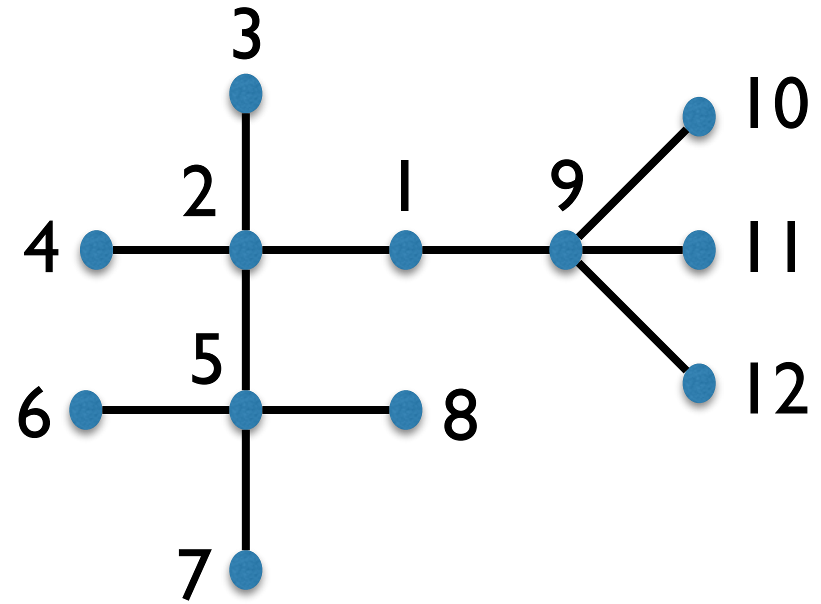






Poll

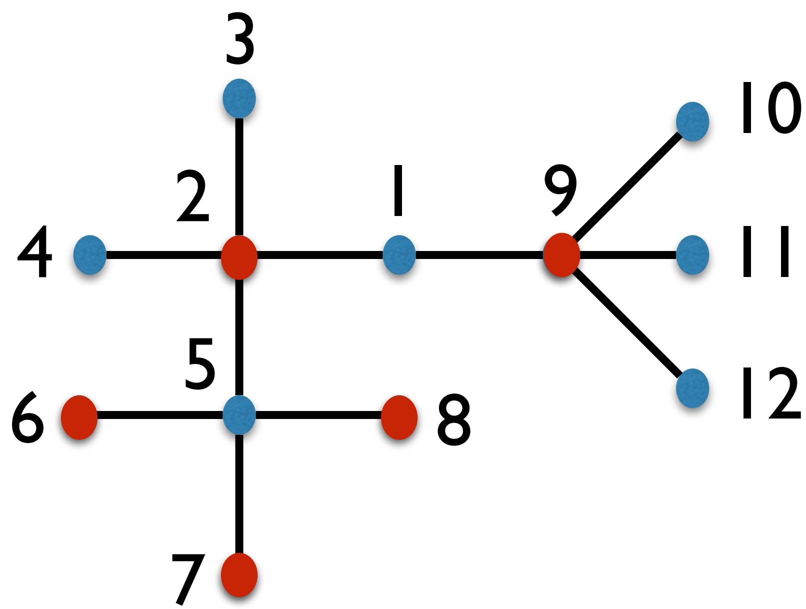
Is this graph bipartite?

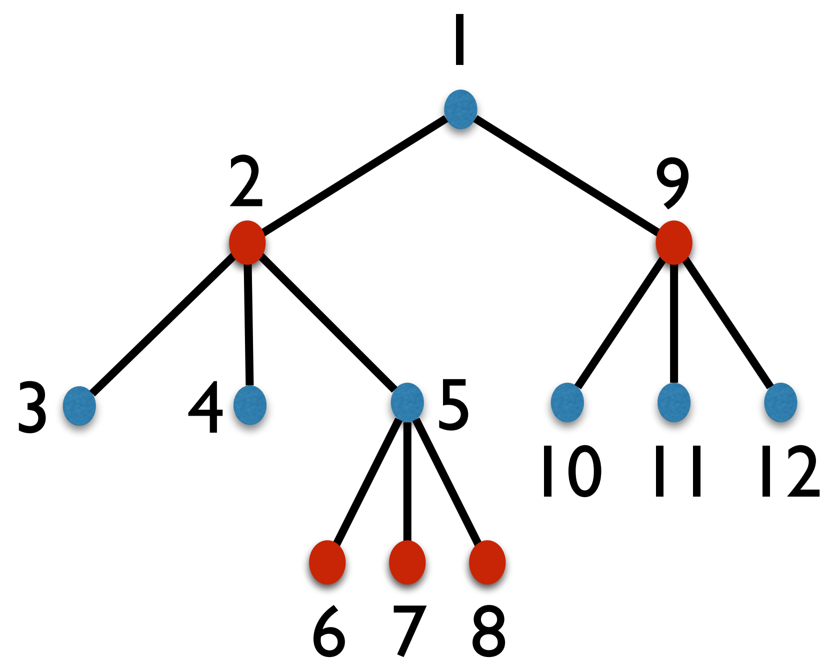


- Yes
- No
- Beats me

Poll Answer

Is this graph bipartite?





bipartite = 2-colorable

Color the vertices with **2 colors** so that no edge's endpoints get the same color.

Important Characterization

<u>An obstruction for being bipartite:</u>

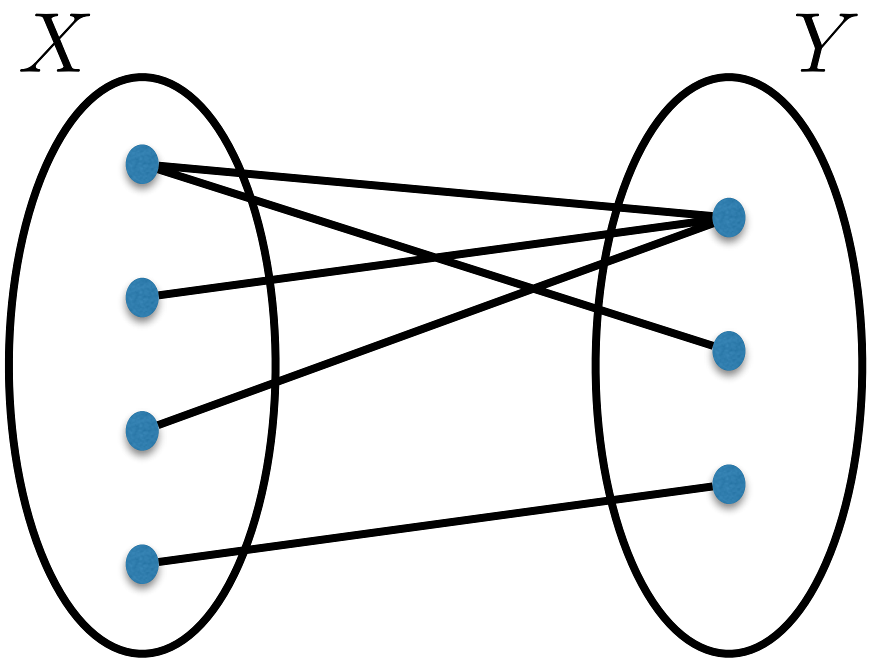
Contains a cycle of odd length.

Is this the only type of obstruction?

<u>Theorem:</u>

A graph G=(V,E) is bipartite **<u>if and only if</u>** it contains no cycles of odd length.

Bipartite Graphs



Often we write the bipartition explicitly:

$$G = (X, Y, E)$$

Bipartite Graphs

Great at modeling relations between two classes of objects.

Examples:

X = machines, Y = jobs

An edge $\{x, y\}$ means x is capable of doing y.

X = professors, Y = courses

An edge $\{x, y\}$ means x can teach y.

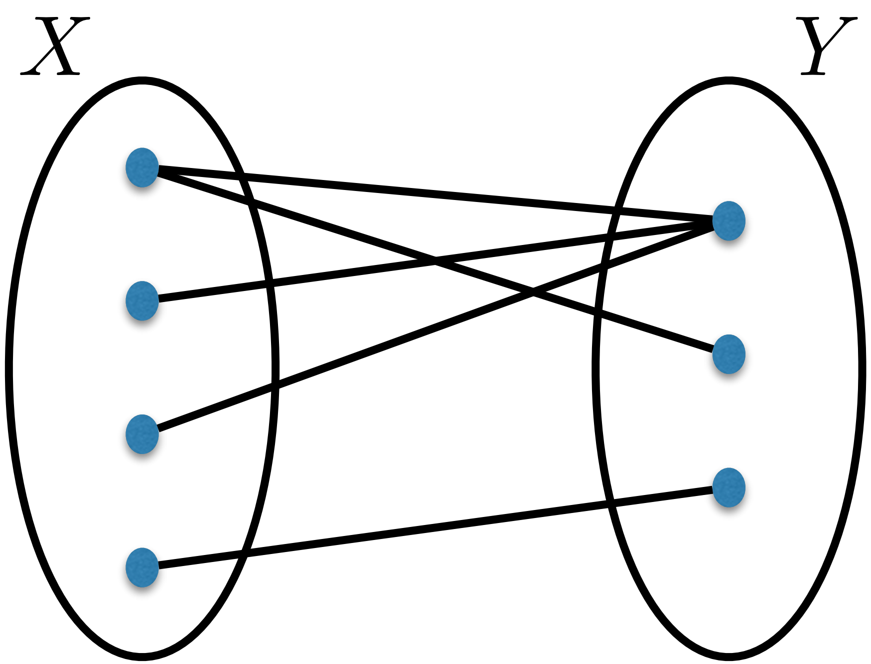
X = students, Y = internship jobs

An edge $\{x, y\}$ means x and y are interested in each other.

$\vdots$

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph

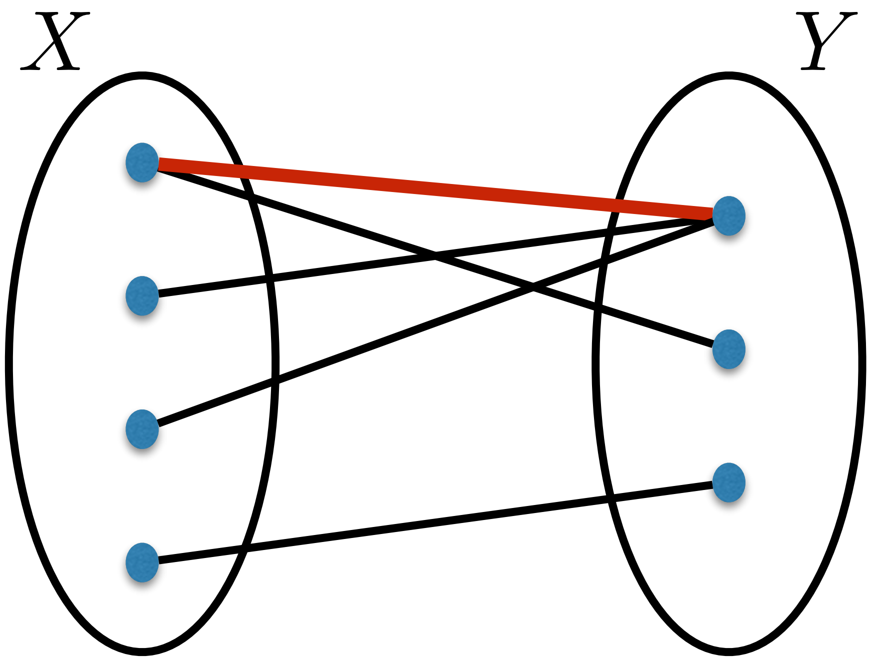


A **matching** :

A <u>subset of the edges</u> that do not share an endpoint.

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph



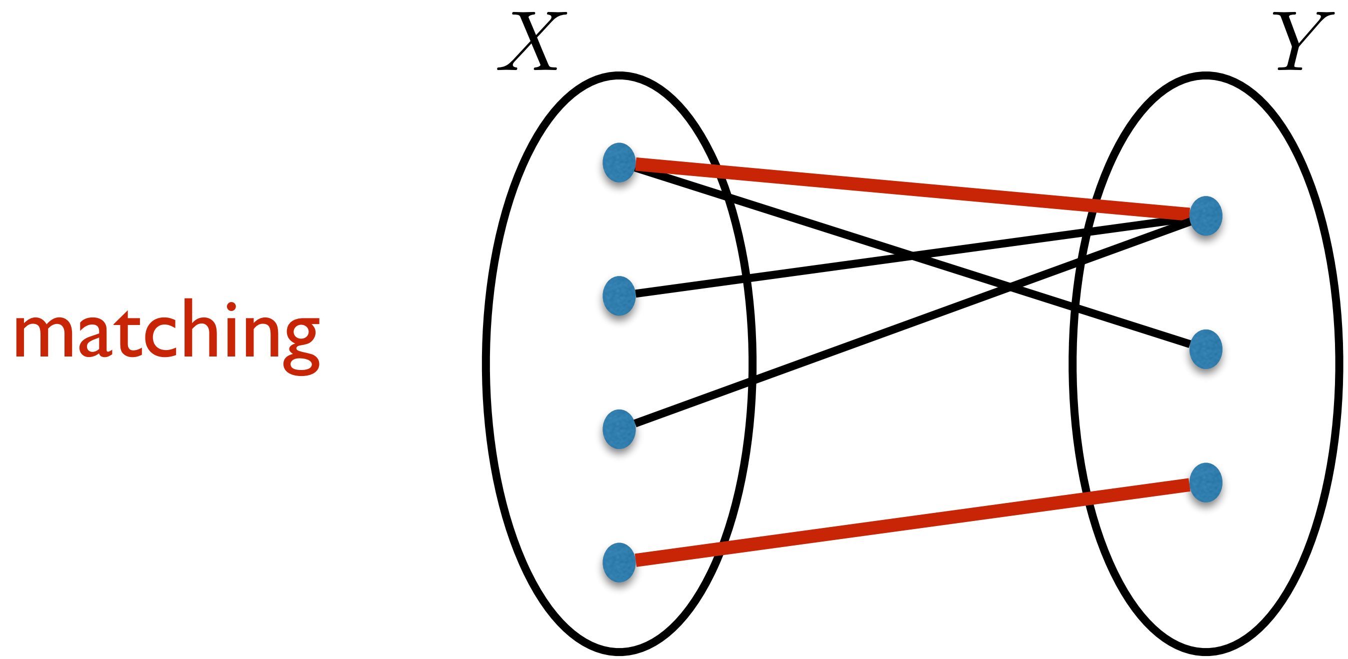
matching

A **matching** :

A <u>subset of the edges</u> that do not share an endpoint.

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph

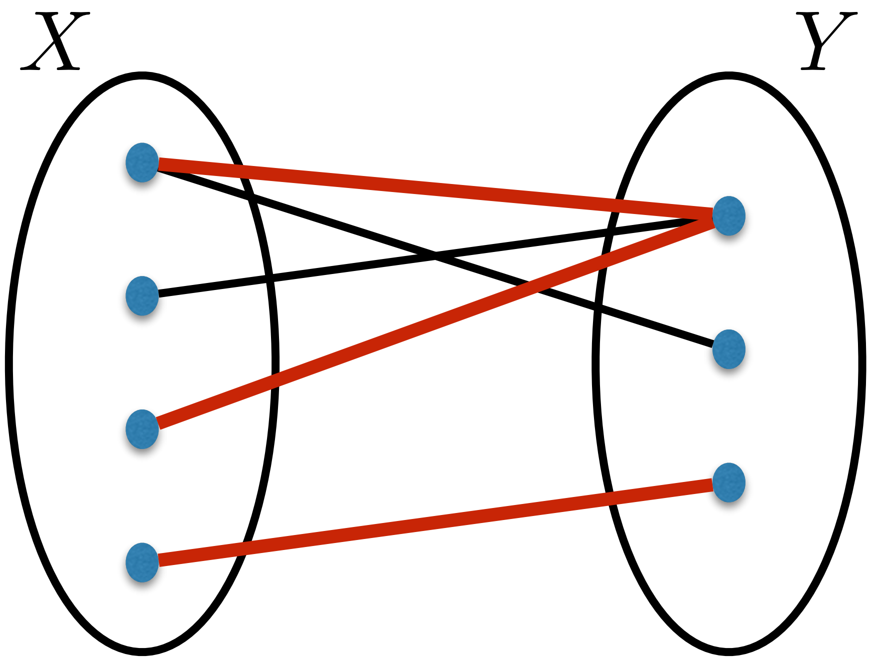


A **matching** :

A <u>subset of the edges</u> that do not share an endpoint.

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph



not a matching

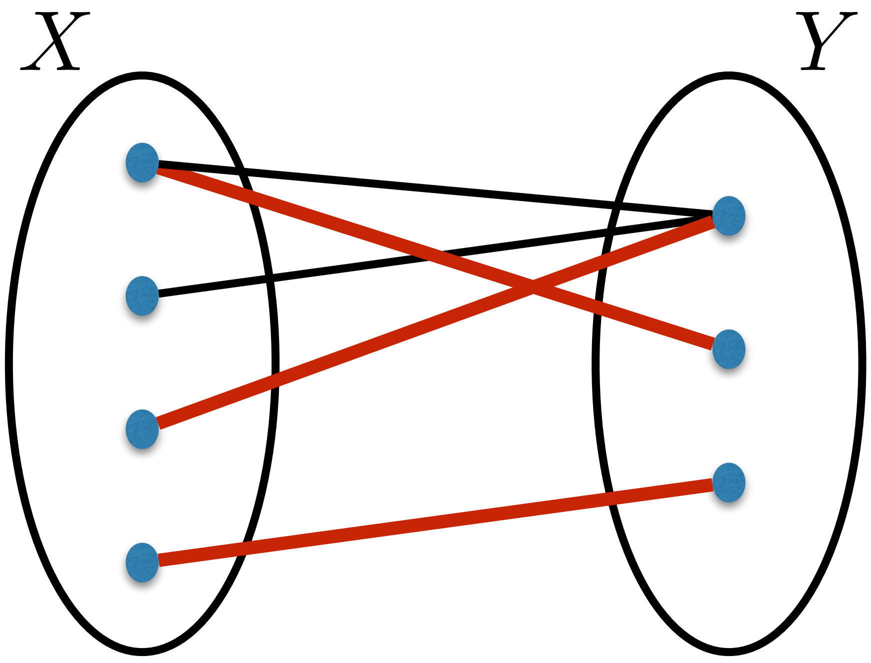
A **matching** :

A subset of the edges that do not share an endpoint.

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph

maximum matching

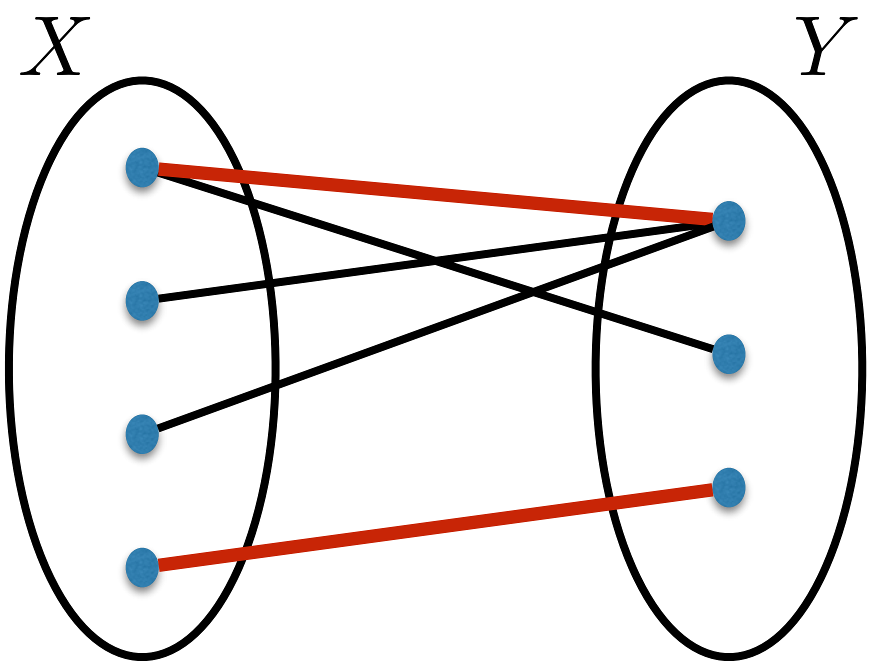


Maximum matching: a matching with largest number of edges (among all possible matchings).

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph

maximal matching



Cannot add more edges.

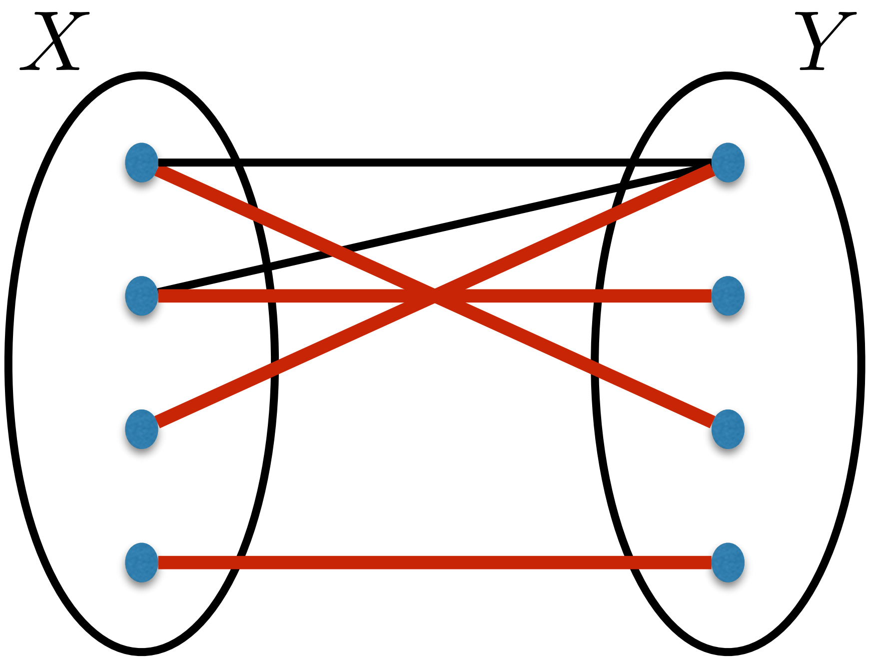
"*local optimum*"

Maximal matching: a matching which cannot contain any more edges.

Matchings in bipartite graphs

Often, we are interested in finding a **matching** in a bipartite graph

perfect matching



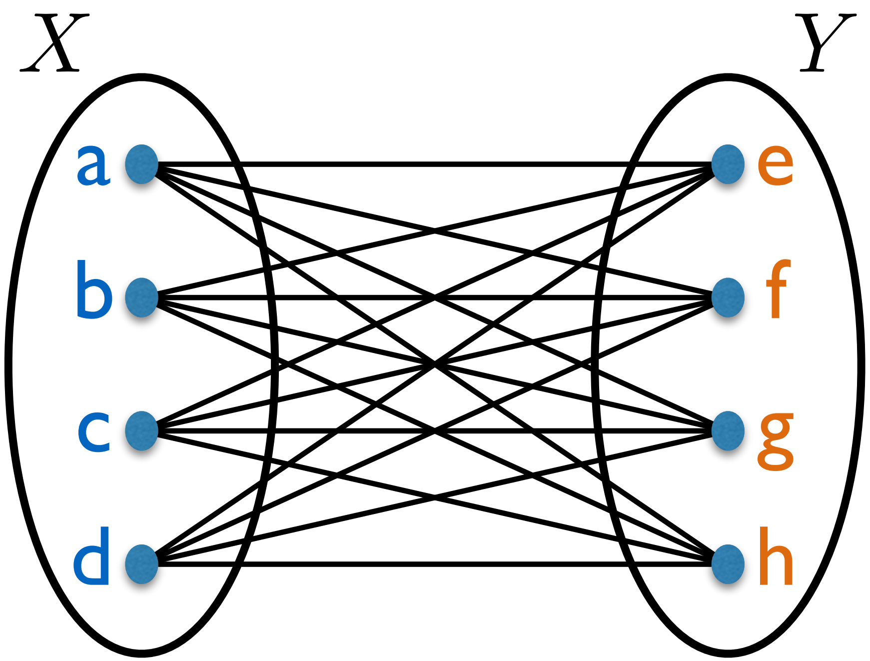
a necessary condition for perfect matching:

$$|X| = |Y|$$

Perfect matching: a matching that covers all vertices.

Poll

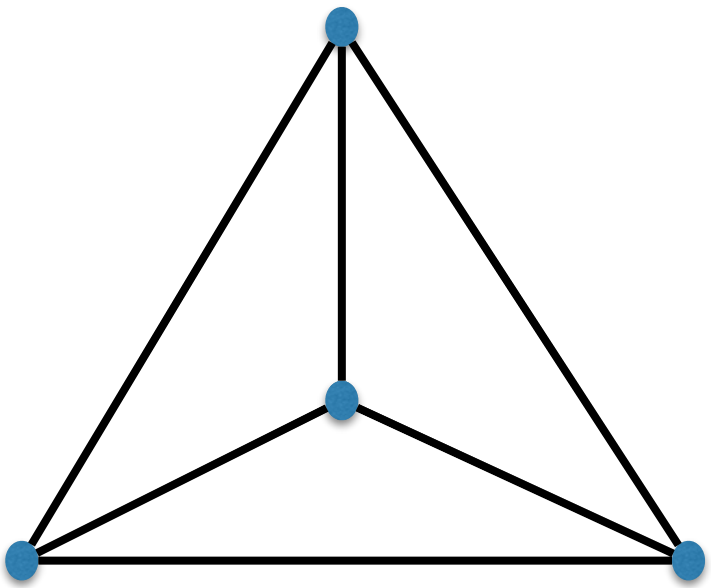
How many different perfect matchings does the graph have (in terms of n)?



$$|X| = |Y| = n$$

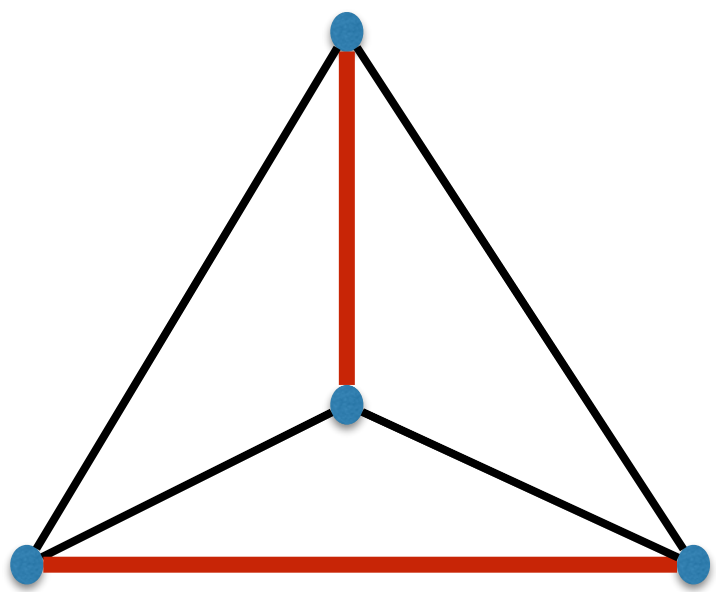
Important Note

We can define matchings for <u>non-bipartite</u> graphs as well.



Important Note

We can define matchings for <u>non-bipartite</u> graphs as well.



Maximum matching problem

The problem we want to solve is:

Maximum matching problem

Input: A graph $G = (V, E)$.

Output: A maximum matching in G.

Bipartite maximum matching problem

Actually, we want to solve the following restriction:

Bipartite maximum matching problem

Input: A *bipartite* graph $G = (X, Y, E)$.

Output: A maximum matching in G.

How do you solve a problem like this?

1. Formulate the problem

2. **Ask**: Is there a trivial algorithm?

3. **Ask**: Is there a better algorithm?

4. Find and analyze

Bipartite maximum matching problem

Bipartite maximum matching problem

Input: A *bipartite* graph $G = (X, Y, E)$.

Output: A maximum matching in G.

Is there a (trivial) algorithm to solve this problem?

- Try all possible subsets of the edges.

Running time: $\Omega(2^m)$

How do you solve a problem like this?

1. Formulate the problem

2. **Ask**: Is there a trivial algorithm?

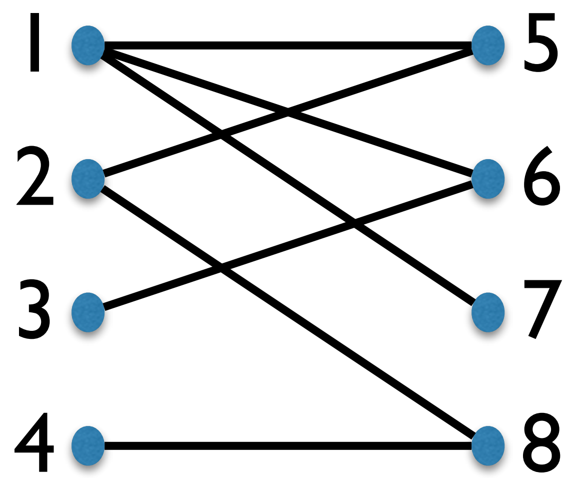
3. **Ask**: Is there a better algorithm?

4. Find and analyze

Bipartite maximum matching problem

A good first attempt:

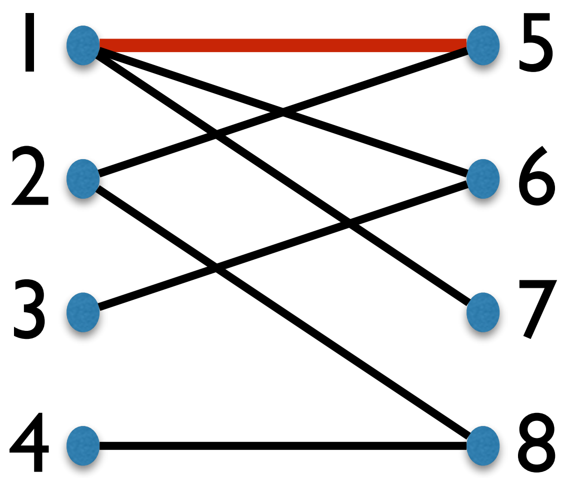
What if we picked edges "greedily"?



Bipartite maximum matching problem

A good first attempt:

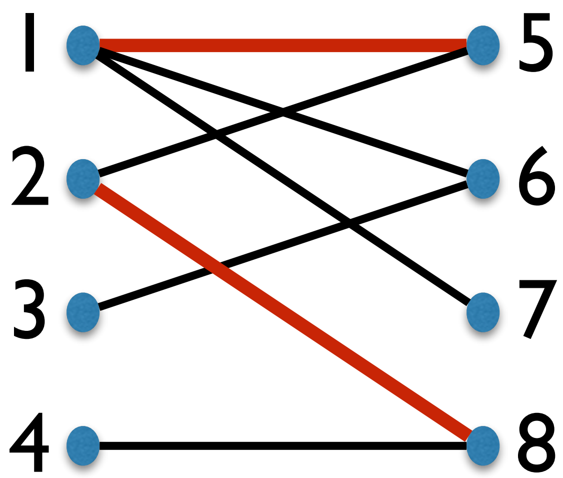
What if we picked edges "greedily"?



Bipartite maximum matching problem

A good first attempt:

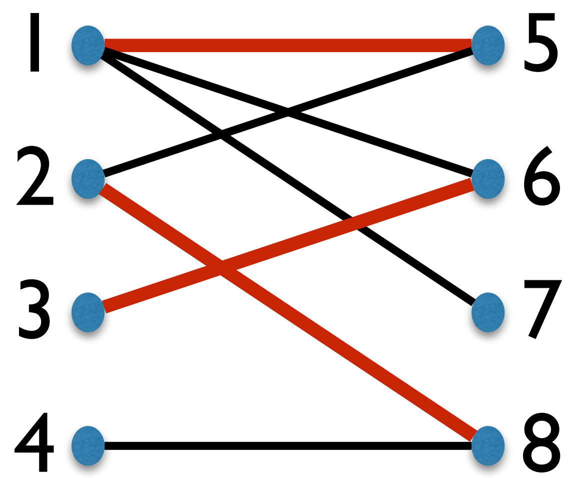
What if we picked edges "greedily"?



Bipartite maximum matching problem

A good first attempt:

What if we picked edges "greedily"?



maximal matching

but not maximum

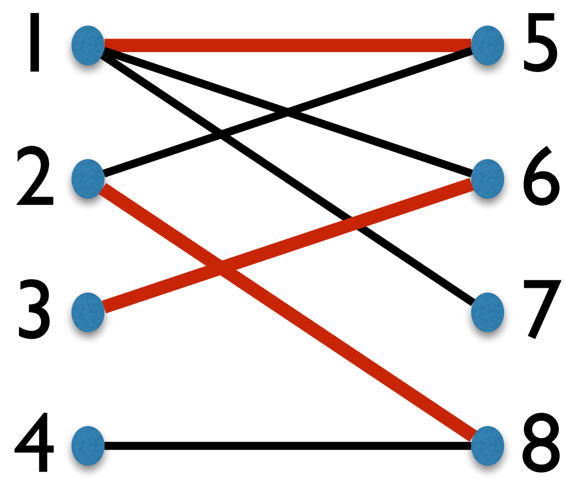
Is there a way to get out of this *local optimum*?

What is interesting about the path 4 – 8 – 2 – 5 – 1 – 7 ?

Bipartite maximum matching problem

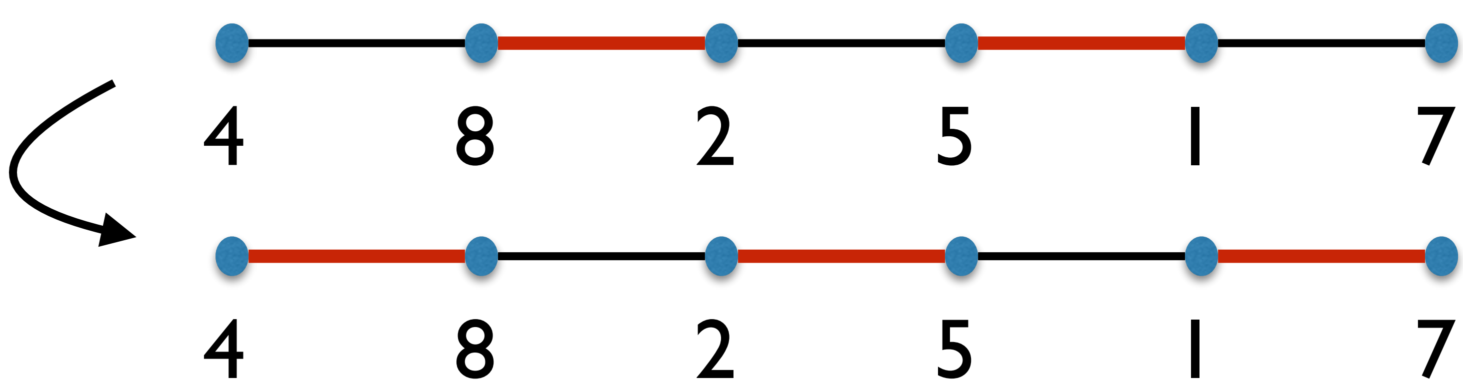
A good first attempt:

What if we picked edges "greedily"?



maximal matching

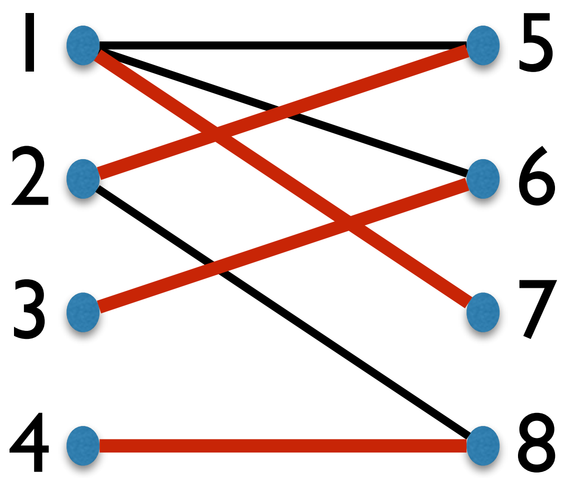
but not maximum



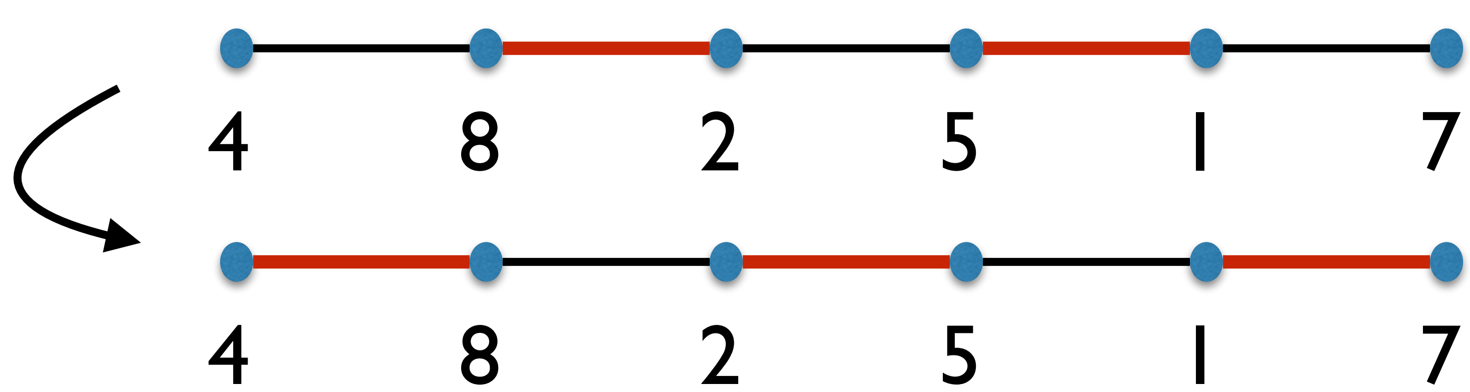
Bipartite maximum matching problem

A good first attempt:

What if we picked edges "greedily"?



now maximum



Important Definition: Augmenting paths

Let M be some matching.

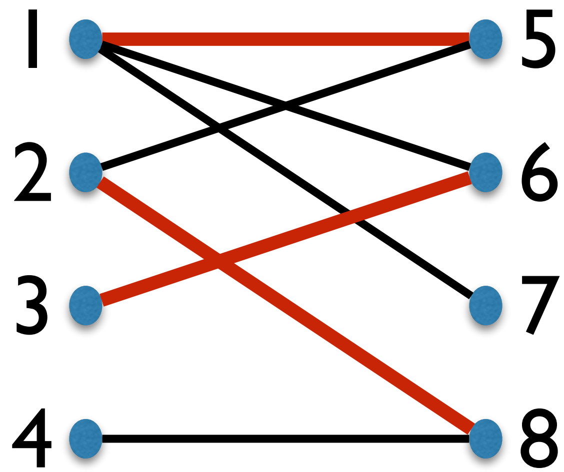
An ***alternating path*** with respect to M is a path in **G** such that:

- the edges in the path alternate between being in M and not being in M

An ***augmenting path*** with respect to M is an alternating path such that:

- the first and last vertices are **not** matched by M

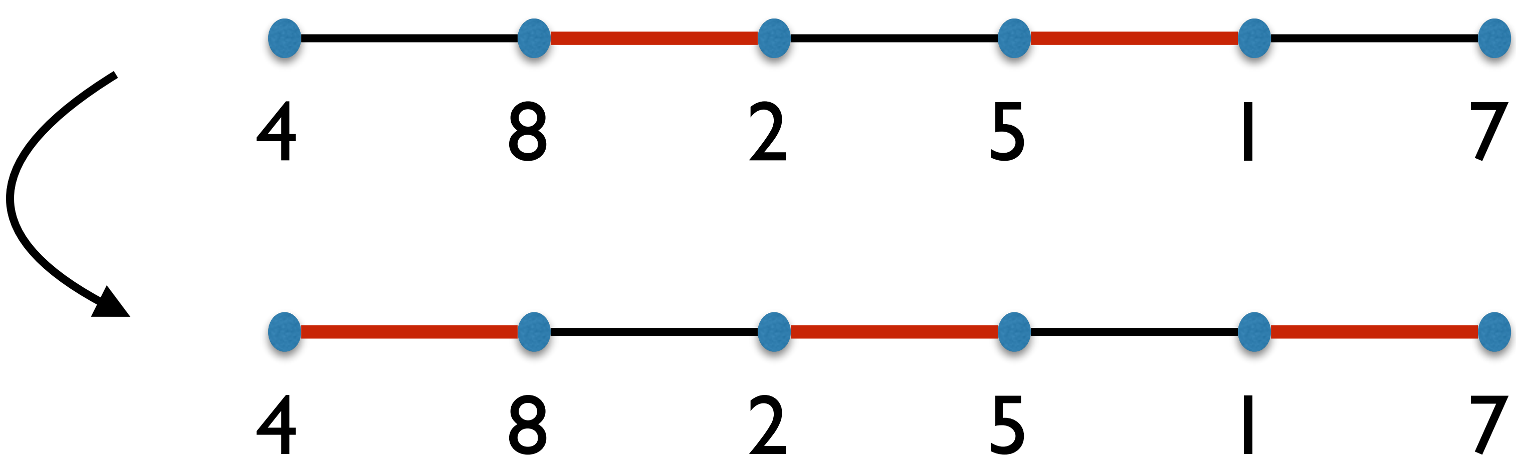
Important Definition: Augmenting paths



matching = red edges

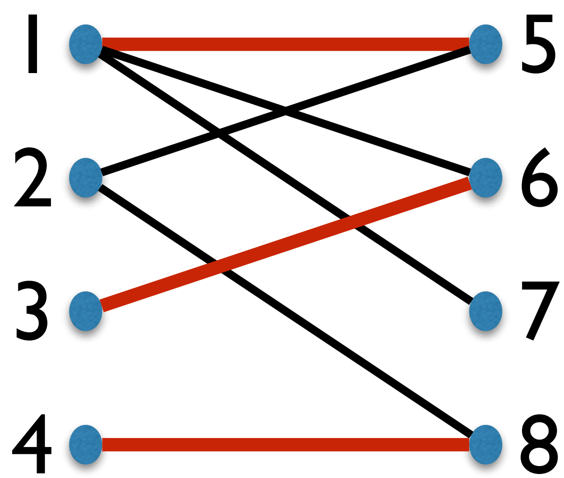
Augmenting path:

4-8-2-5-1-7



augmenting path $\Longrightarrow$ can obtain a bigger matching.

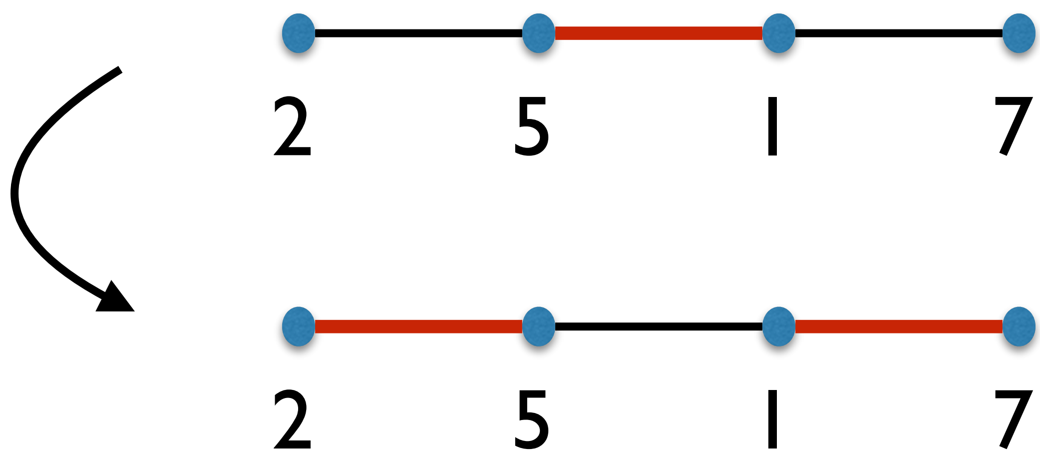
Important Definition: Augmenting paths



matching = red edges

Augmenting path:

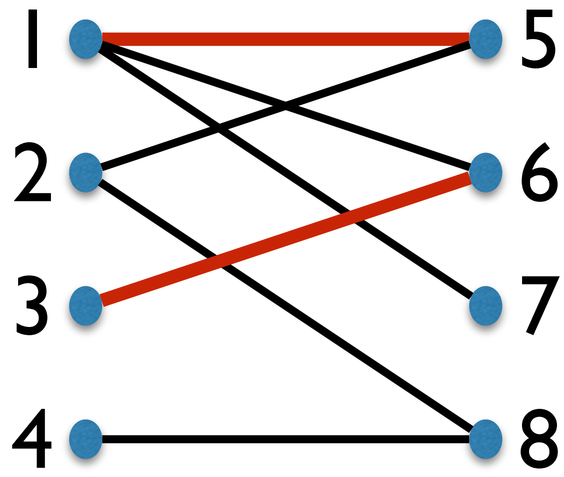
2-5-1-7



An augmenting path
need **not** contain
all the edges of the matching.

augmenting path $\Longrightarrow$ can obtain a bigger matching.

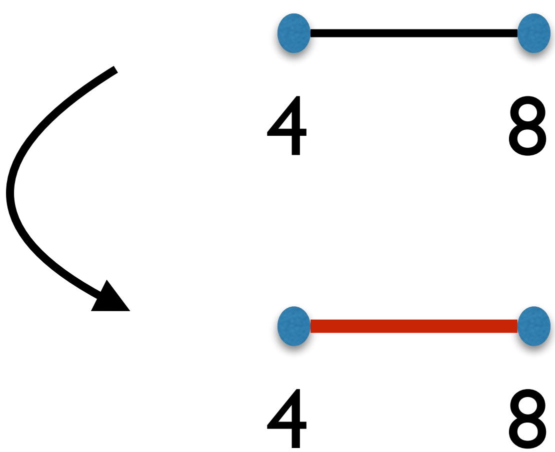
Important Definition: Augmenting paths



matching = red edges

Augmenting path:

4-8



An augmenting path
need **not** contain
any of the edges of the matching.

augmenting path $\Longrightarrow$ can obtain a bigger matching.

Augmenting paths and maximum matchings

augmenting path $\Longrightarrow$ can obtain a bigger matching.

In fact, it turns out:

no augmenting path $\Longrightarrow$ maximum matching.

Theorem:

A matching **M** is maximum **if and only if** there is **no** augmenting path with respect to **M**.

Augmenting paths and maximum matchings

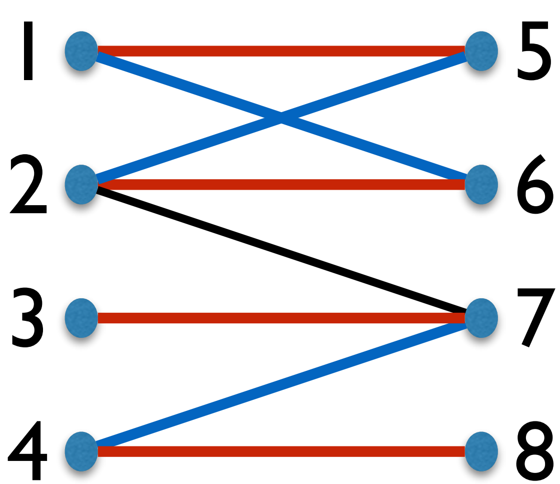
Proof:

If there is an augmenting path with respect to **M**,
we saw that **M** is not maximum.

Want to show:

If **M** not maximum, there is an augmenting path w.r.t. **M**.

Let **M*** be a maximum matching. |**M***| > |**M**|.

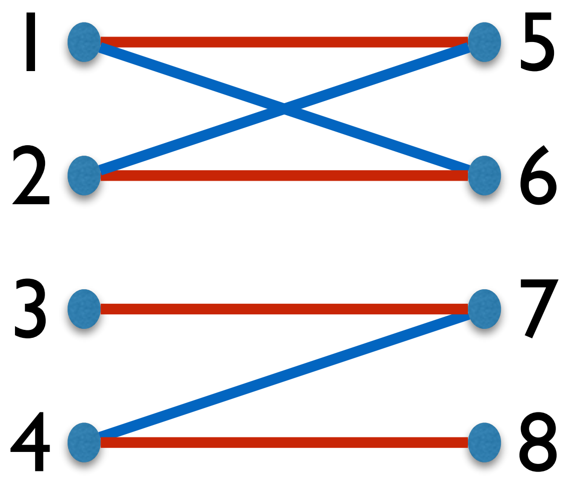


Let **S** be the set of edges contained in **M*** or **M** but not both.

$S = (M^* \cup M) - (M \cap M^*)$

Augmenting paths and maximum matchings

Proof (continued):



Let **S** be the set of edges contained in **M*** or **M** but not both.

S = (**M*** $\cup$ **M**) - (**M** $\cap$ **M***)

(will find an augmenting path in **S**)

What does **S** look like?

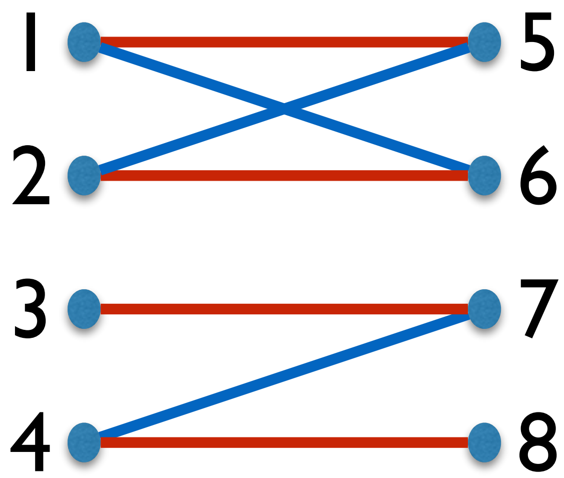
Each vertex has degree 1 or 2. (why?)

So **S** is a collection of disjoint **cycles** and **paths**. (exercise)

The edges alternate **red** and **blue**.

Augmenting paths and maximum matchings

Proof (continued):



Let **S** be the set of edges contained in **M*** or **M** but not both.

S = (**M*** $\cup$ **M**) - (**M** $\cap$ **M***)

So **S** is a collection of disjoint **cycles** and **paths**.

The edges alternate **red** and **blue**.

\# red > # blue in **S**

\# red = # blue in **cycles**

So $\exists$ a **path** with # red > # blue.

This is an ***augmenting path*** with respect to **M**. $\square$

Augmenting paths and maximum matchings

Theorem:

A matching **M** is maximum **if and only if** there is **no** augmenting path with respect to **M**.

Summary of proof:

$\Longrightarrow$

If there is an augmenting path, not a max matching.

$\Longleftarrow$

If the matching **M** is not maximum, $\exists$ **M*** s.t. |**M***| > |**M**|.

Can find an augmenting path w.r.t. **M** in the "symmetric difference" of **M*** and **M**.

Next time:

- Algorithm to find a maximum matching in bipartite graphs.

- Stable matchings.

Questions about the midterm exam