Online Algorithms

Performance-based Advertising

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

Online Algorithms

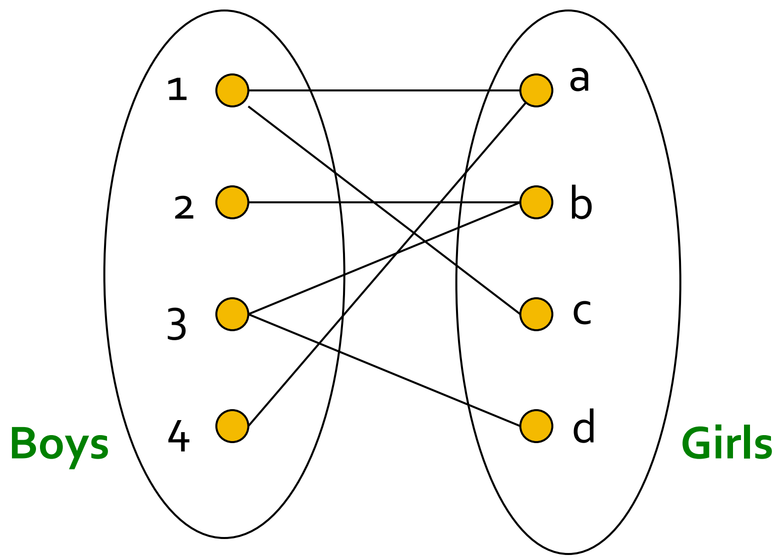
- **Classic model of algorithms**
  - You get to see the entire input, then compute some function of it
  - In this context, "offline algorithm"
- **Online Algorithms**
  - You get to see the input one piece at a time, and need to make irrevocable decisions along the way
  - **Similar to the data stream model**

Online Algorithms

Bipartite Matching

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

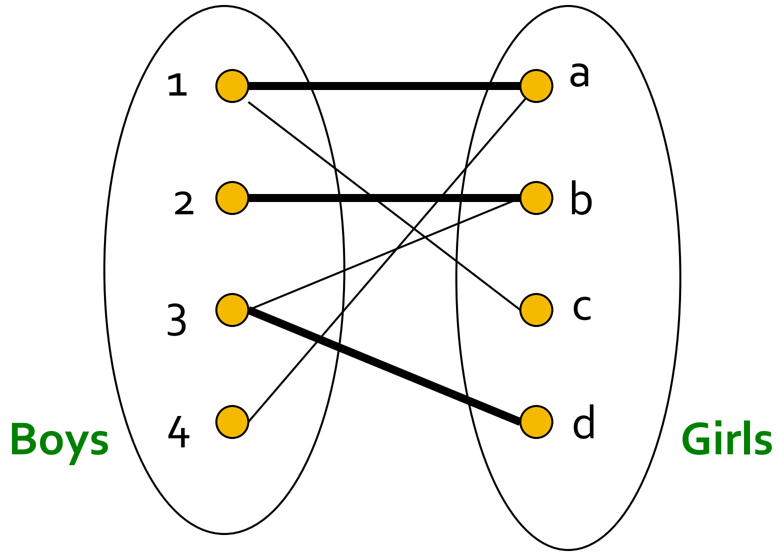
Example: Bipartite Matching



Nodes: Boys and Girls; Edges: Compatible Pairs

Goal: Match as many compatible pairs as possible

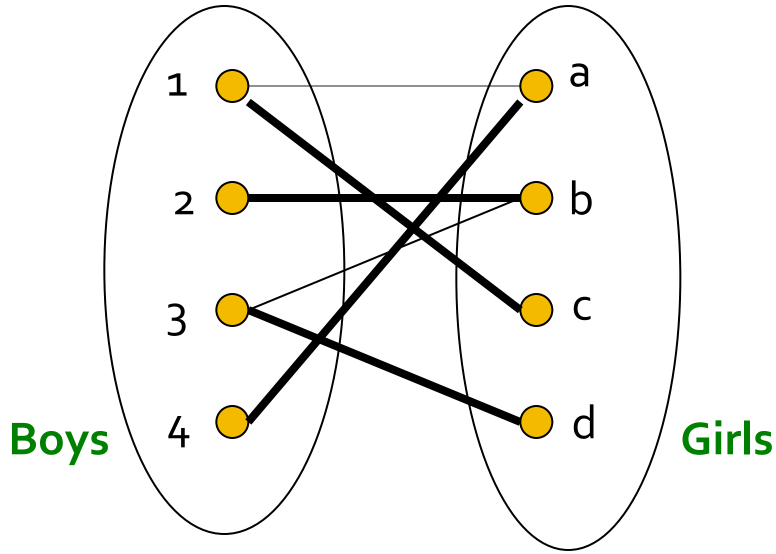
Example: Bipartite Matching



M = {(1,a),(2,b),(3,d)} is a **matching**

Cardinality of matching = |M| = 3

Example: Bipartite Matching



M = {(1,c),(2,b),(3,d),(4,a)} is a **perfect matching**

Perfect matching … all vertices of the graph are matched
Maximum matching … a matching that contains the largest possible number of matches

Matching Algorithm

- **Problem: Find a maximum matching for a given bipartite graph**
  - A perfect one if it exists

- There is a polynomial-time offline algorithm based on augmenting paths (Hopcroft & Karp 1973, see http://en.wikipedia.org/wiki/Hopcroft-Karp_algorithm)

- **But what if we do not know the entire graph upfront?**

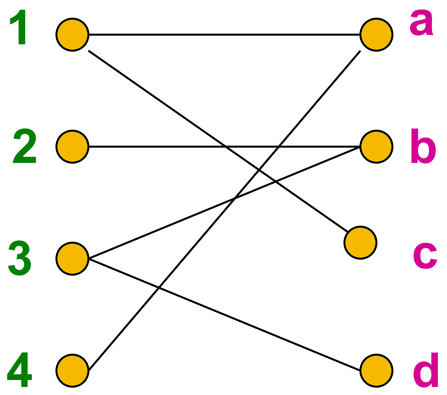
Online Graph Matching Problem

- Initially, we are given the set **boys**
- In each **round**, **one girl's choices are revealed**
  - That is, girl's **edges** are revealed
- **At that time, we have to decide to either:**
  - Pair the **girl** with a **boy**
  - Do not pair the **girl** with any **boy**

- **Example of application:**

  Assigning tasks to servers

Online Graph Matching: Example



(1,a)
(2,b)
(3,d)

Greedy Algorithm

- **Greedy algorithm for the online graph matching problem:**
  - Pair the new girl with **any** eligible boy
    - If there is none, do not pair girl
- **How good is the algorithm?**

Competitive Ratio

- For input ***I***, suppose greedy produces matching $\boldsymbol{M_{greedy}}$ while an optimal matching is $\boldsymbol{M_{opt}}$

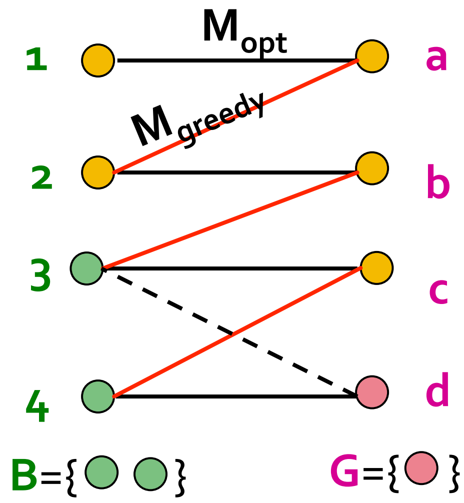
Competitive ratio =

$$\min_{\text{all possible inputs } I} (|M_{greedy}|/|M_{opt}|)$$

(what is greedy's <u>worst</u> performance <u>over all possible</u> inputs *I*)

Analyzing the Greedy Algorithm

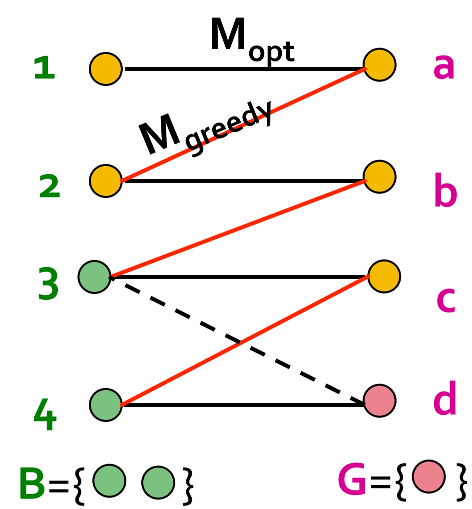
- Suppose $M_{greedy} \neq M_{opt}$
- Consider the set G of girls matched in M_{opt} but not in M_{greedy}
- (1) $|M_{opt}| \leq |M_{greedy}| + |G|$



- Every boy B <u>adjacent</u> to girls in G is already matched in M_{greedy}
- (2) $|M_{greedy}| \geq |B|$

Analyzing the Greedy Algorithm

- **So far:**
  - *G* matched in M_{opt} but not in M_{greedy}
  - Boys *B* adjacent to girls *G*
  - (1) $|M_{opt}| \leq |M_{greedy}| + |G|$
  - (2) $|M_{greedy}| \geq |B|$

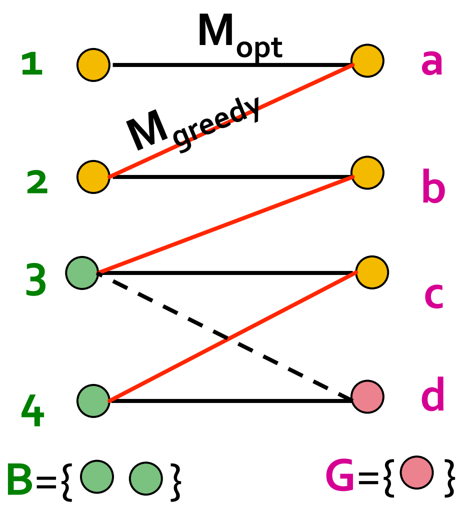


- Optimal matches all the girls in *G* to boys in *B*
  - (3) $|G| \leq |B|$

- Combining (2) and (3):
  - (4) $|G| \leq |B| \leq |M_{greedy}|$

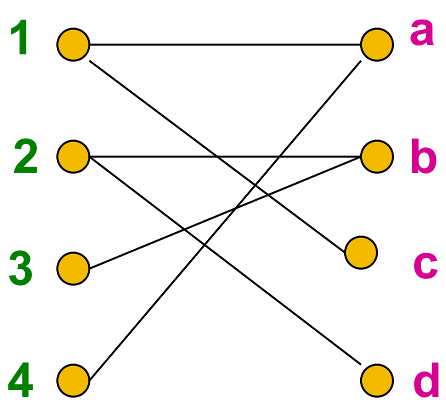
Analyzing the Greedy Algorithm

- So we have:
  - (1) $|\mathbf{M}_{\mathbf{opt}}| \leq |\mathbf{M}_{\mathbf{greedy}}| + |\mathbf{G}|$
  - (4) $|\boldsymbol{G}| \leq |\mathbf{B}| \leq |\boldsymbol{M}_{\boldsymbol{greedy}}|$



- Combining (1) and (4):
  - $|\mathbf{M}_{\mathbf{opt}}| \leq |\mathbf{M}_{\mathbf{greedy}}| + |\mathbf{M}_{\mathbf{greedy}}|$
  - $|\mathbf{M}_{\mathbf{opt}}| \leq 2|\mathbf{M}_{\mathbf{greedy}}|$
  - $|\boldsymbol{M}_{\boldsymbol{greedy}}|/|\boldsymbol{M}_{\boldsymbol{opt}}| \geq \mathbf{1/2}$

Worst-case Scenario



(1,a)
(2,b)