

Graph Matching

Instructor: Meng-Fen Chiang

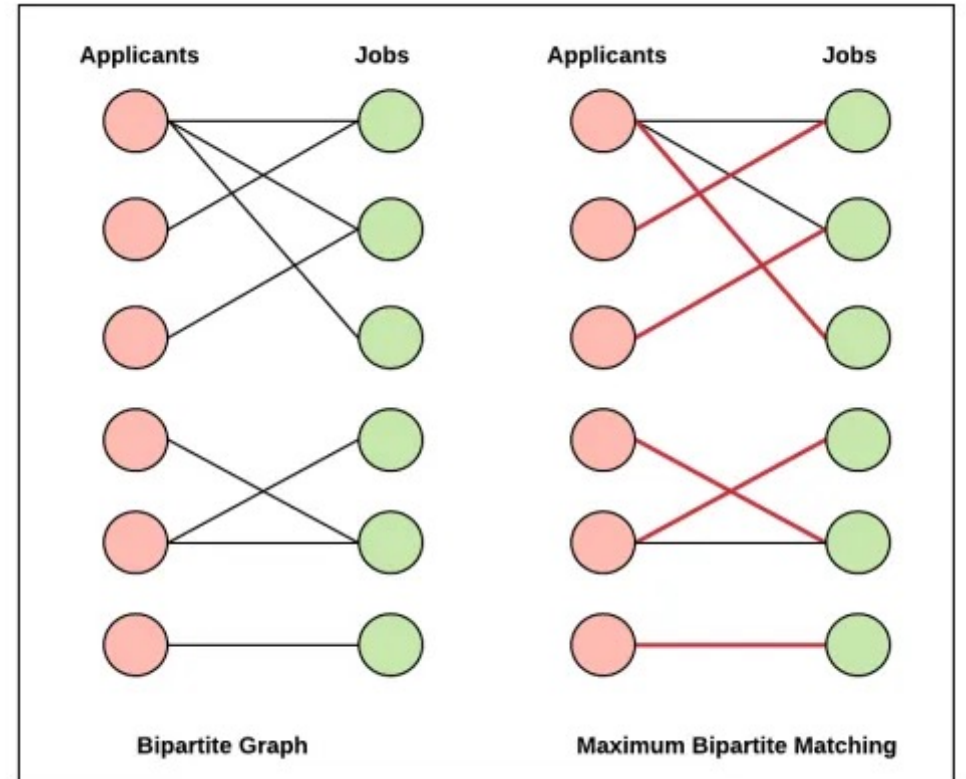
COMPCSI220: WEEK 12



Slides adapted from Mark Wilson, Georgy Gimel'farb, Simone Linz and Tanya Gvozdeva

OUTLINE

- Matching Problem
- Maximal and Maximum Matching
- Alternating Paths
- Augmenting Paths



Matching Problem

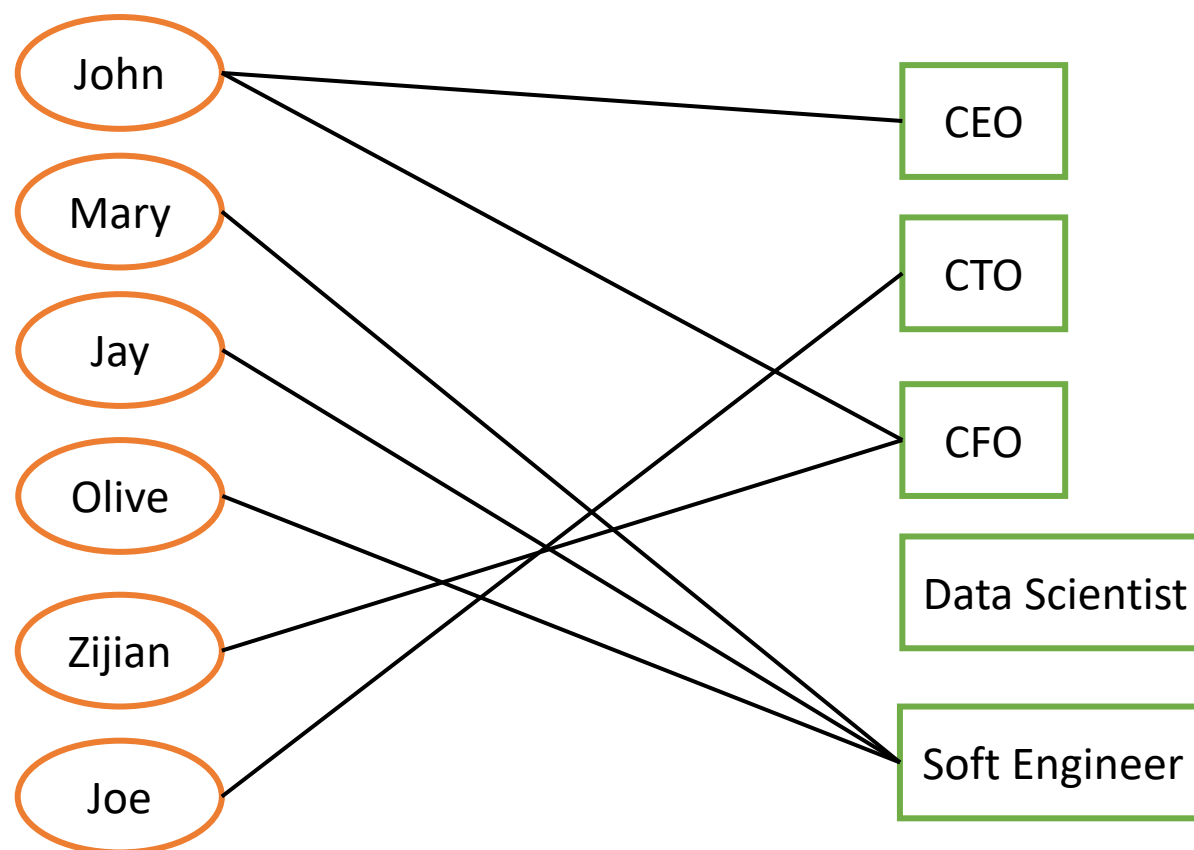
Team

- John
- Mark
- Jay
- Olive
- Zijian
- Joe

Roles

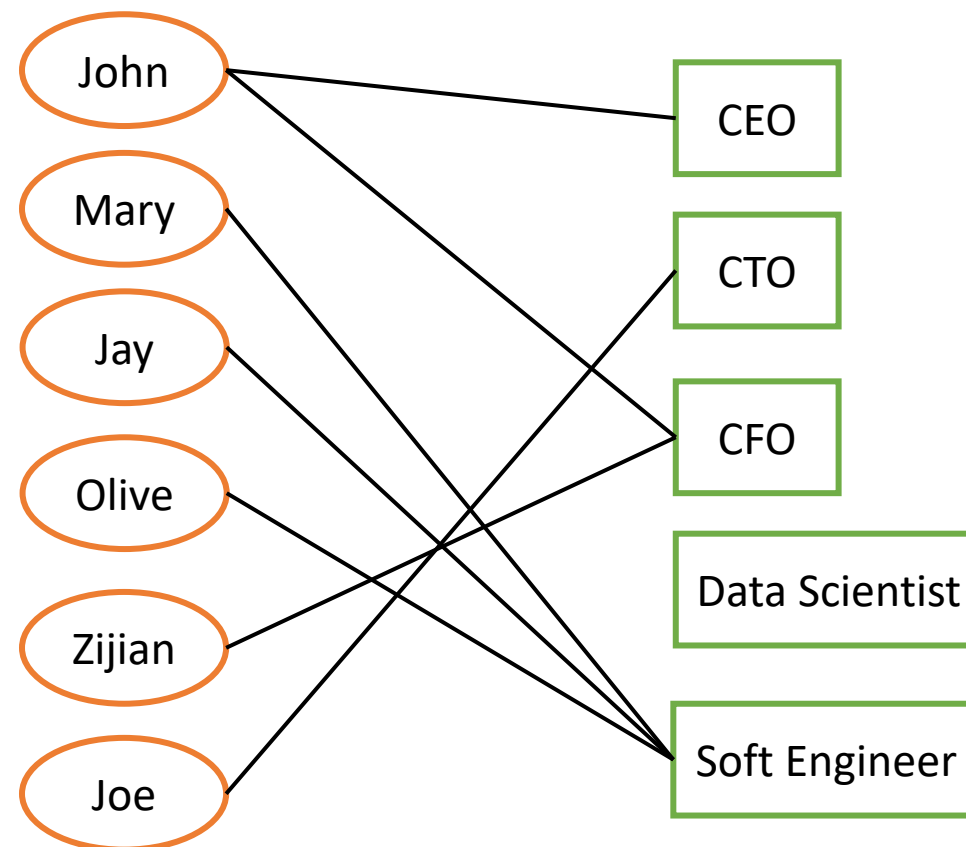
- CEO
- CTO
- CFO
- Data Scientist
- Soft Engineer

Reducing to a Graph Problem



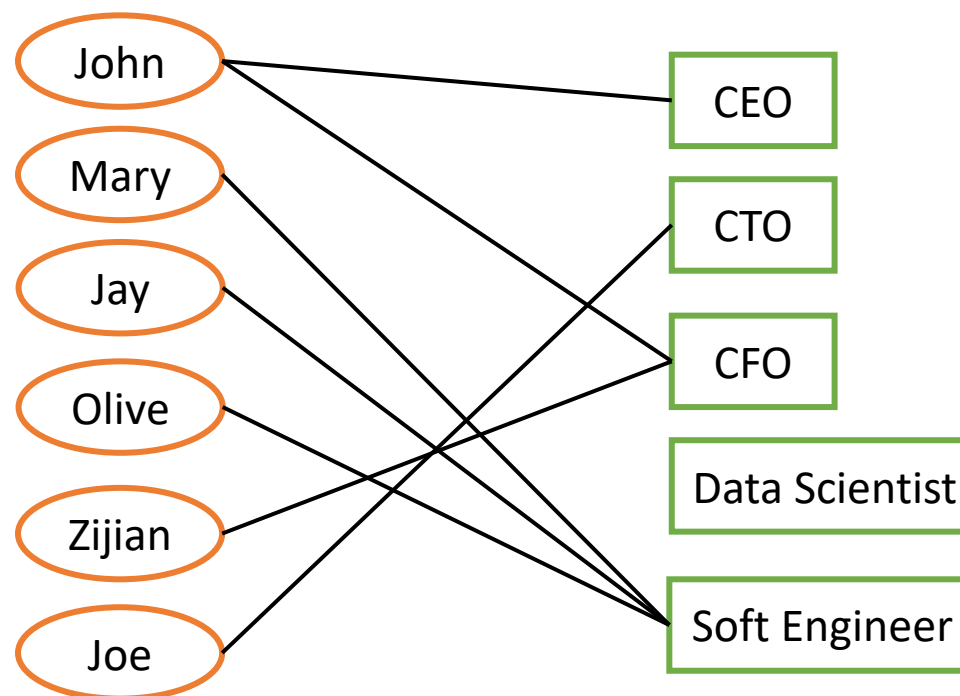
Bi-partite Graphs

- A graph $G(V, E)$ is bi-partite if
 - $V(G)$ can be partitioned in two non-empty disjoint subsets $\{v_0, v_1\}$ s.t.
 - Each edge of $E(G)$ connects one vertex of v_0 and one vertex of v_1



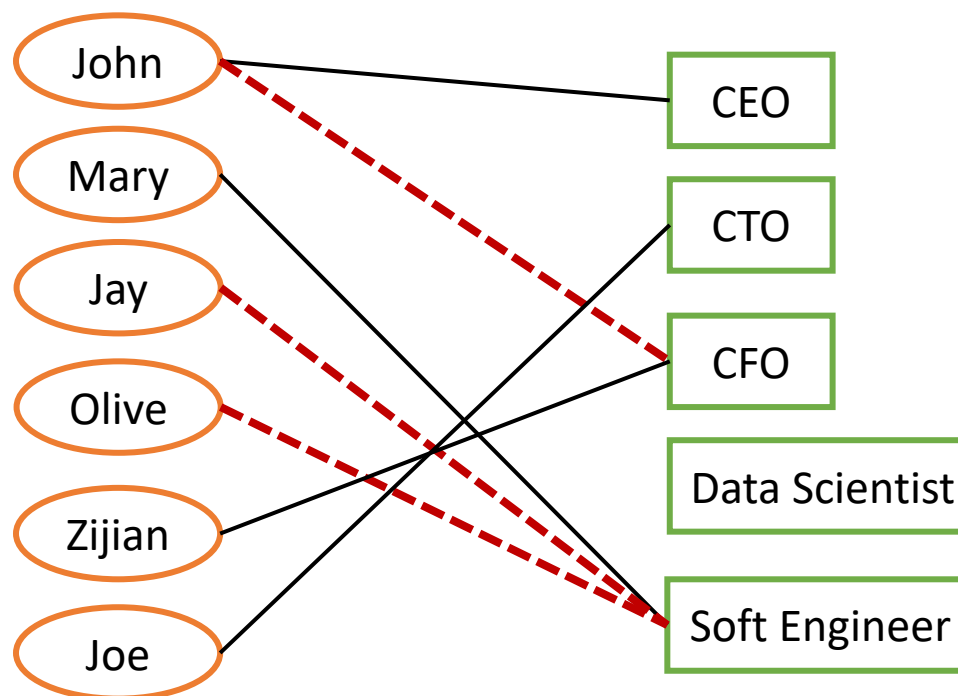
Matching

- We cannot have more than one person for the key roles: CEO, CTO and CFO



Matching (Contd.)

- In other words, we want to find edges that do not start/end on the same vertices

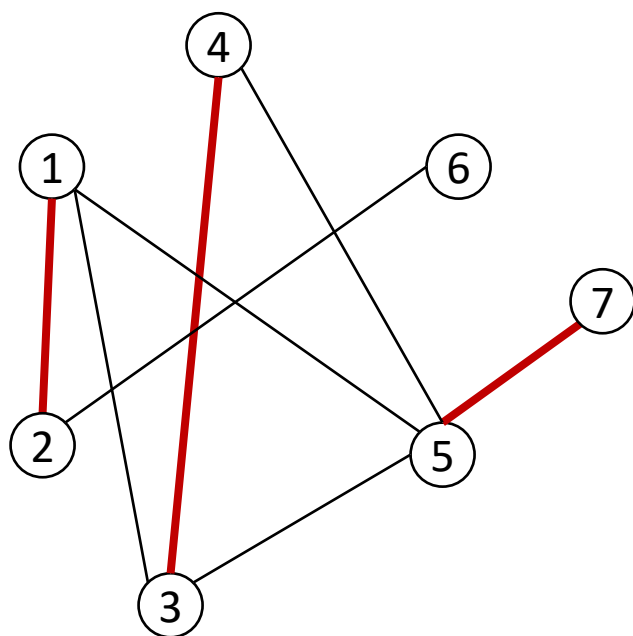


Matchings

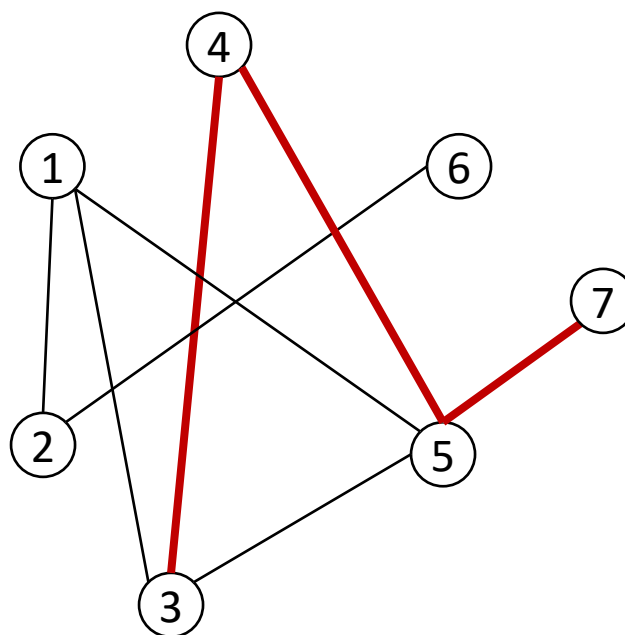
- A **matching** M is a set of pairwise non-adjacent edges in a graph

Example: Matchings

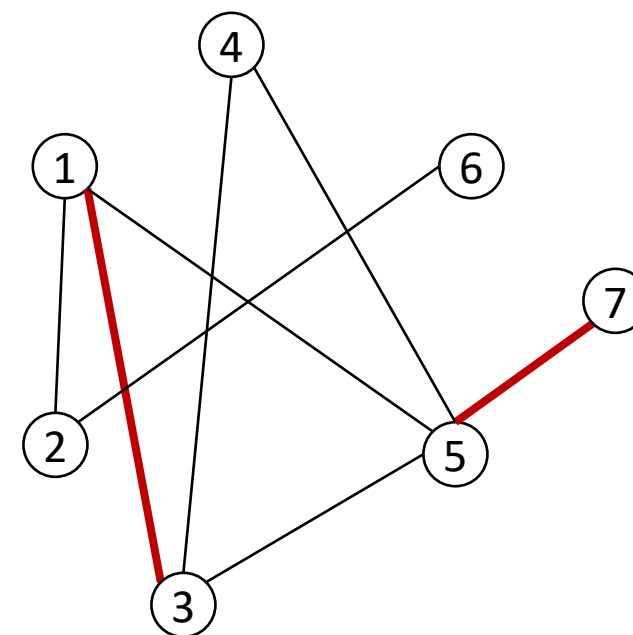
- Is this a matching?



YES



NO

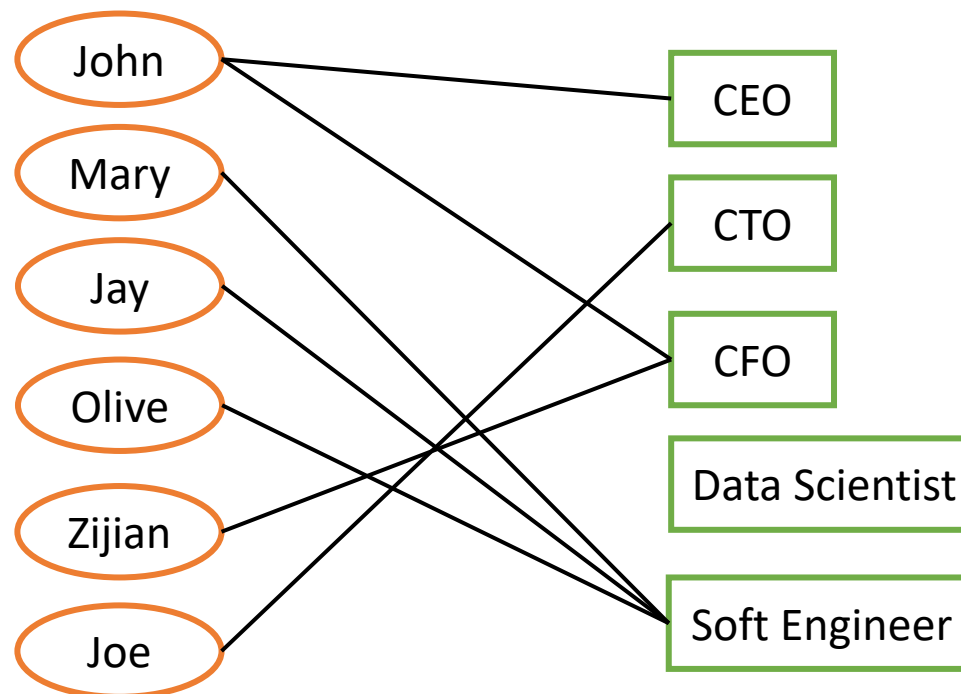


YES

Importance of Matchings

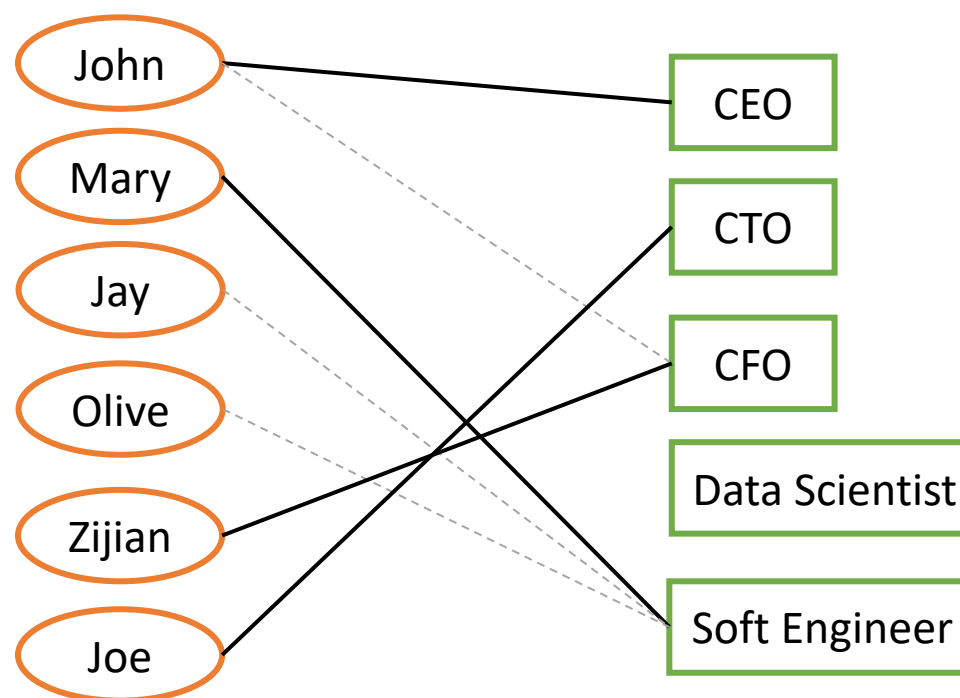
- Matchings are used whenever we need to assign members of a set to each other as exclusive pairs based on suitability criteria
- The members of the set are the vertices, and the edges indicate potentially suitable pairings
- Matchings are often but not always used in conjunction with bipartite graphs

This is a Possible Matching

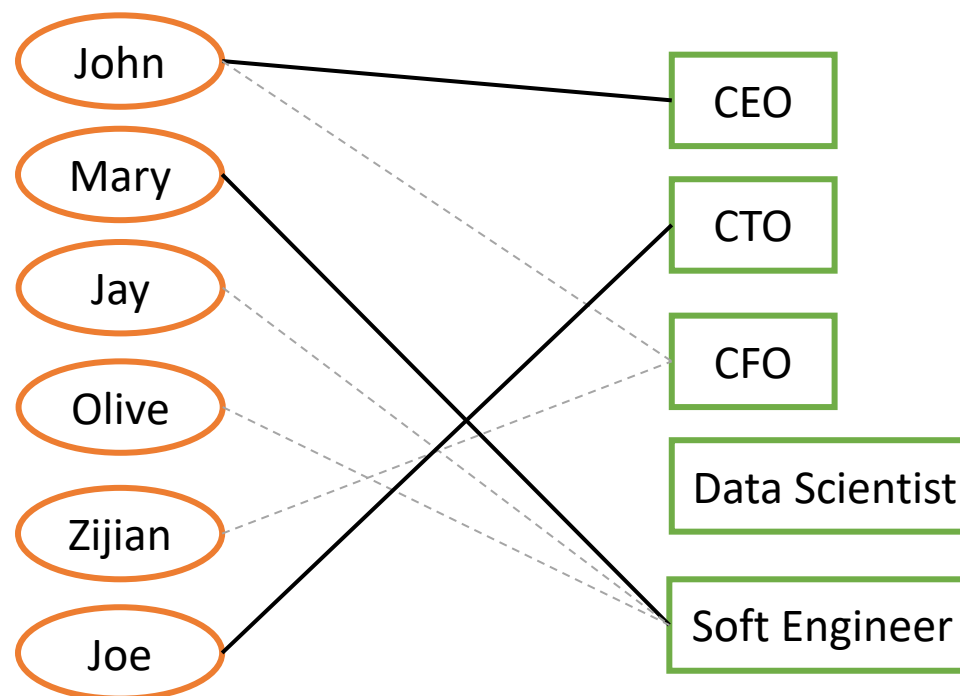


Maximal Matching

- It is not a subset of any other matching



Maximal Matching (Contd.)



Maximal and Maximum Matchings

- A **maximal** matching is a matching that is **not a subset of any other matching**.
- A **maximum** matching is a matching with the **highest number of edges**.
- In other words, we cannot find a matching with more edges

Maximal and Maximum Matchings

- A maximum matching is always a maximal matching, but a maximal matching is not necessarily a maximum matching.
- Note: A maximum matching in an arbitrary graph G can be found in polynomial time. For a polynomial time algorithm see section 5.9 in the textbook.

Exercise: Car Ferry

- A small car ferry across a river can take two vehicles at a time with a combined weight of 4,000 kg. On one side, the following cars are waiting for a trip across:

- Please Find

- All edges between vertices that respect the constraint of 4000Kg
- An example of maximal matching
- An example of maximum matching

BMW
2,300kg

Holden
1,800kg

Fiat
1,650kg

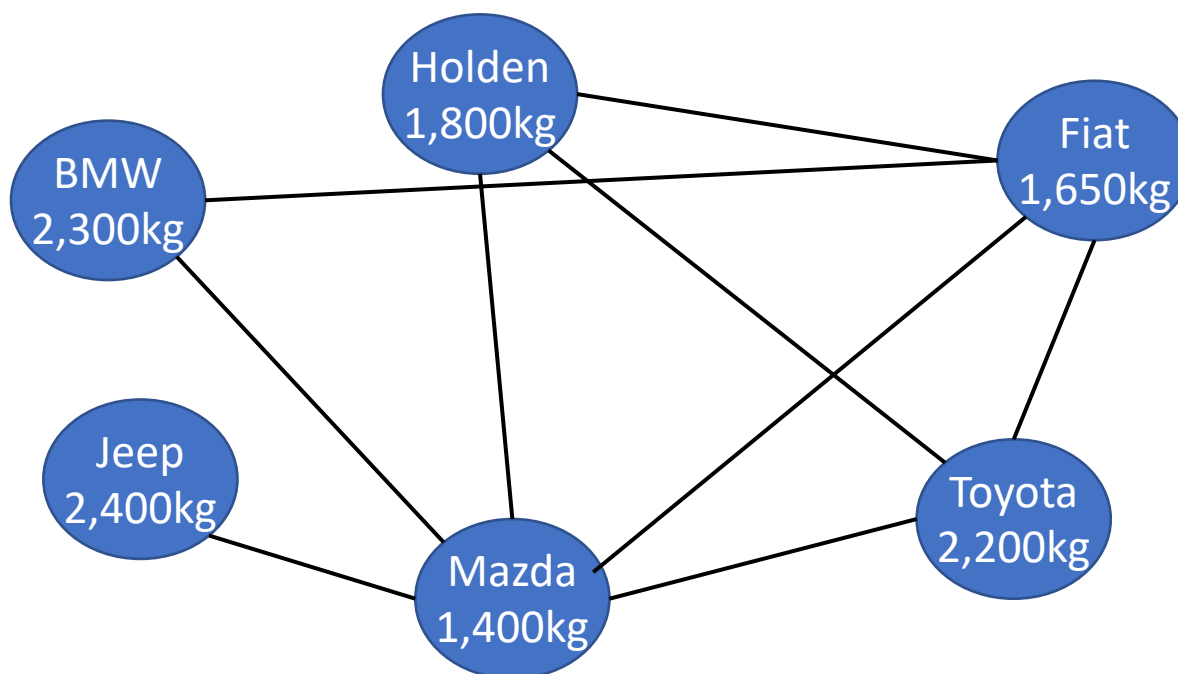
Jeep
2,400kg

Mazda
1,400kg

Toyota
2,200kg

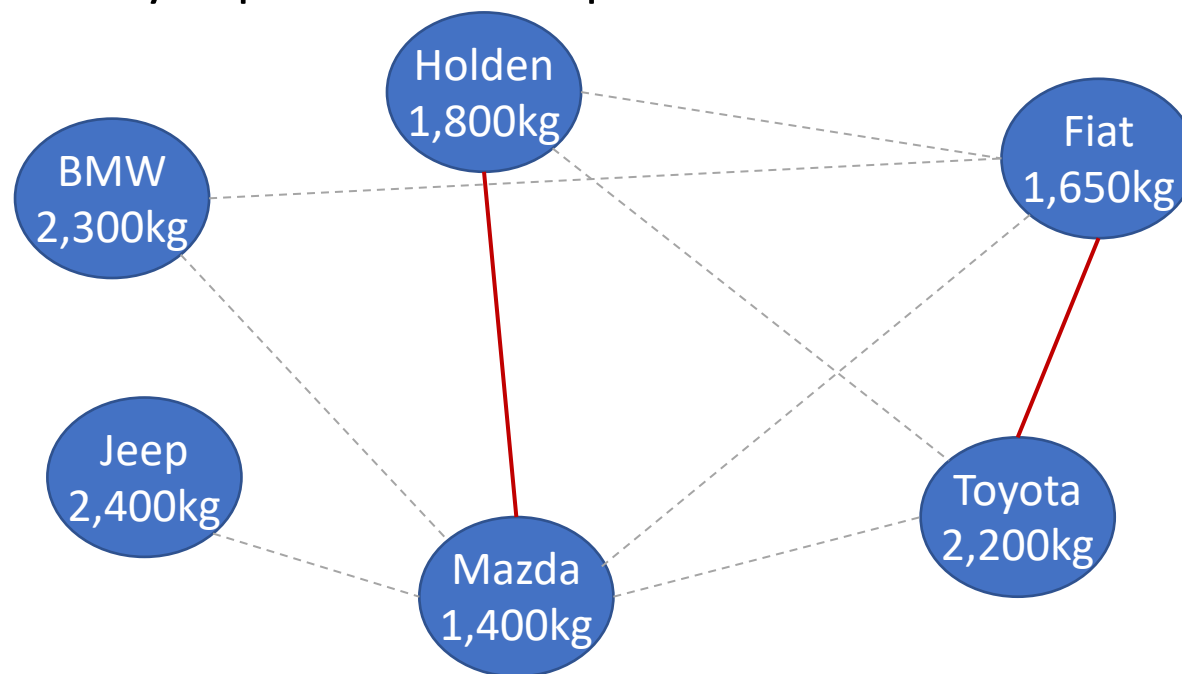
Exercise: Car Ferry

- Edges indicate cars that can be put on the ferry together:



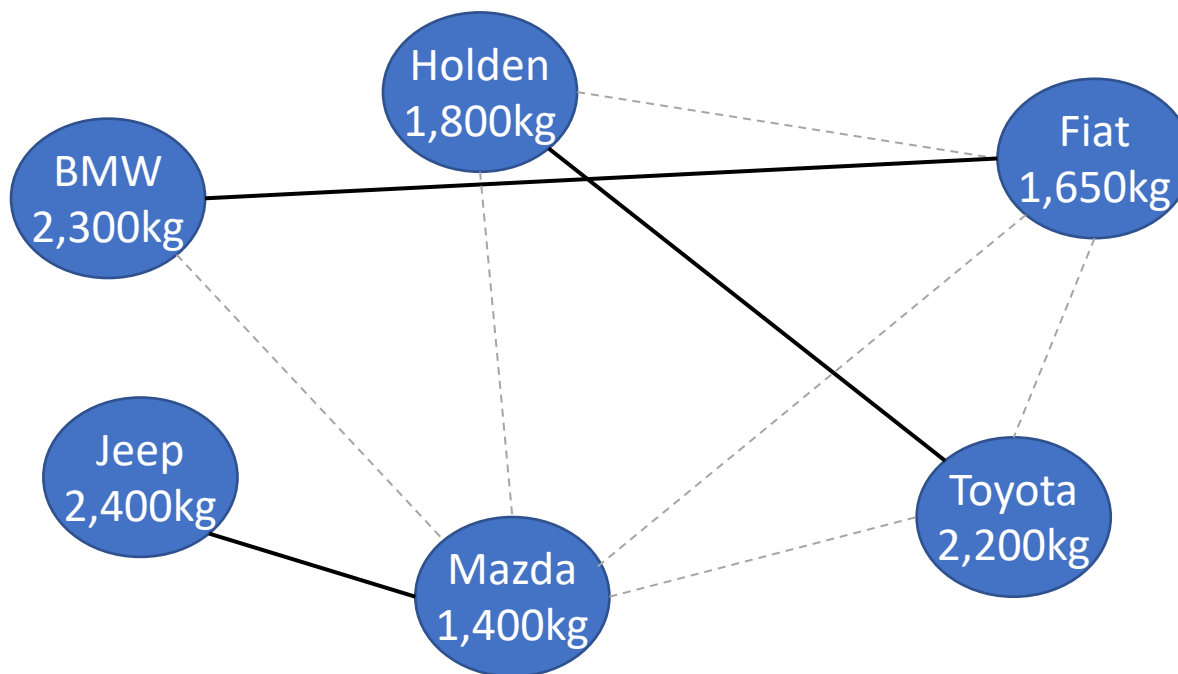
Exercise: Car Ferry

- A matching indicates a possible way in which to get the cars across
- We require four ferry trips: two with a pair of cars and two with a single car each.



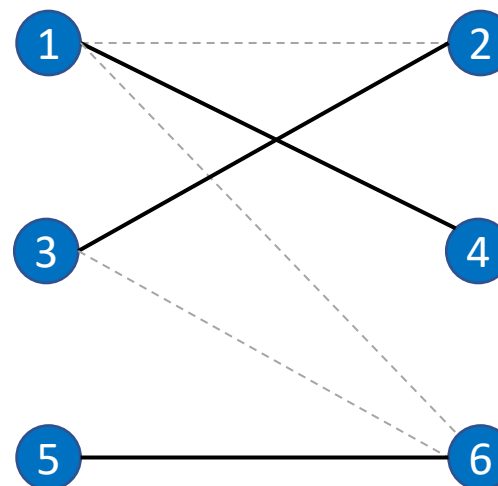
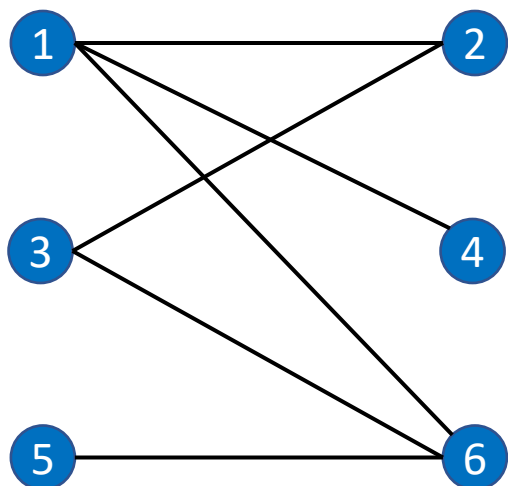
Exercise: Car Ferry

- In this matching, we can get all cars across in three ferry trips:



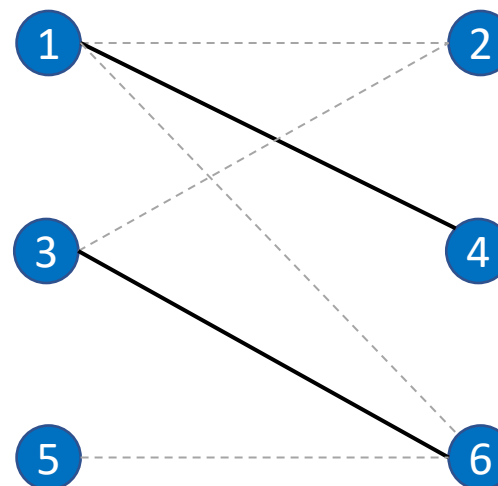
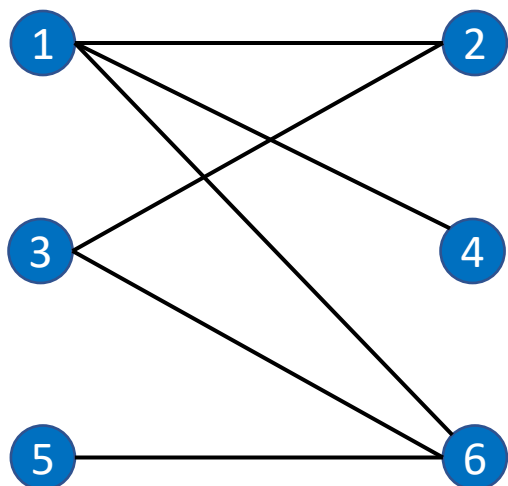
Finding Maximum Matchings

- A maximum matching in an arbitrary graph G can be found in polynomial time
- Given a graph $G(V,E)$, a **matching** M in G is a set of pairwise non-adjacent edges
 - No two edges share a common vertex



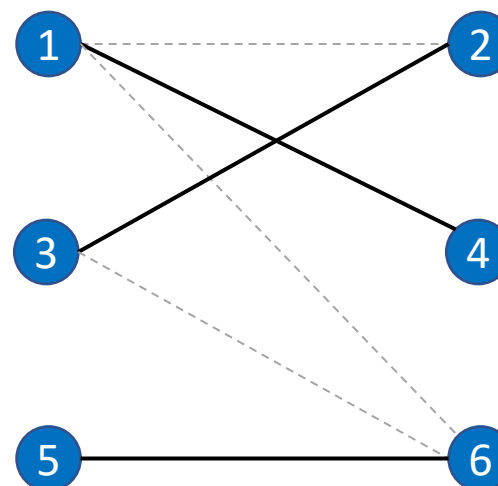
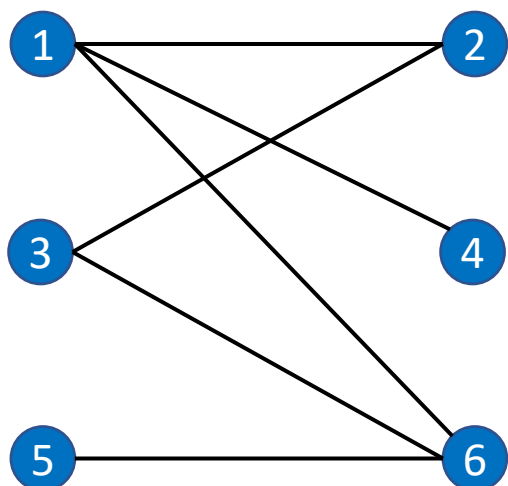
Finding Maximum Matchings (Contd.)

- Given a graph $G(V,E)$, a matching M is a **maximal matching** if after adding another edge to M , M is not a matching any longer



Finding Maximum Matchings (Contd.)

- Given a graph $G(V,E)$, a matching M is a **maximum matching** if it contains the largest number of edges



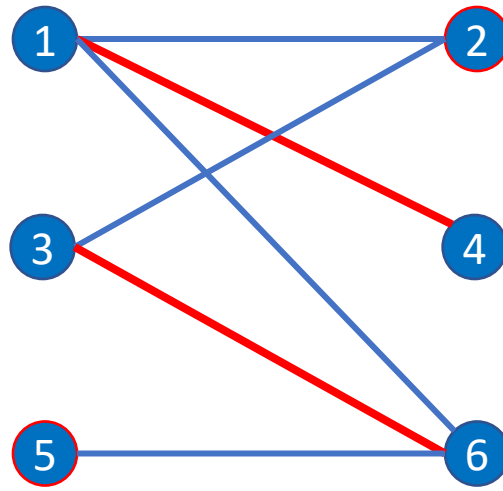
Finding Maximal and Maximum Matching

- A simple greedy algorithm that searches maximal matchings:
 - Iterates over all edges in the graph
 - Add an edge to a maximal matching if it is not adjacent to any other edges in the matching

(Un-)Matched Vertex

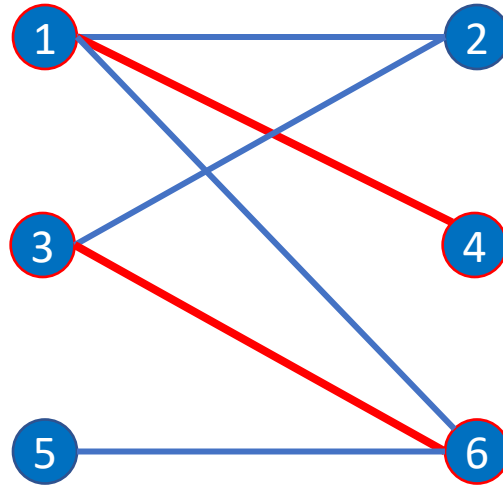
- A vertex is **matched** (or **saturated**) if it is an endpoint of one of the edges in the matching
- A vertex is **unmatched** if no edges of the matching end/start from such vertex

Example: (Un-)Matched Vertex



2 and 5 are unmatched

Example: (Un-)Matched Vertex



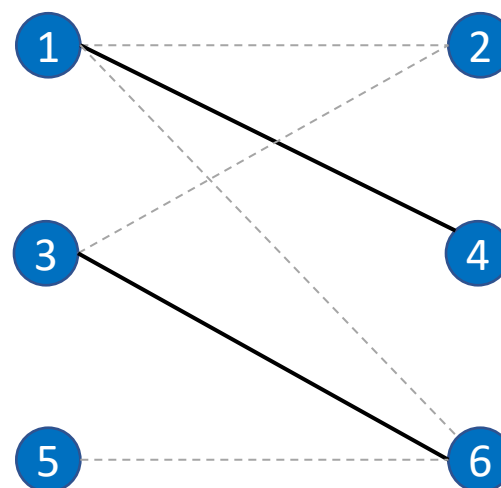
1,4,3,6 are matched

Alternating Paths

- An **alternating path** in a matching M is a path **starting from an unmatched vertex** in which the **edges alternate** from being in M and not

Path: 2,1,4

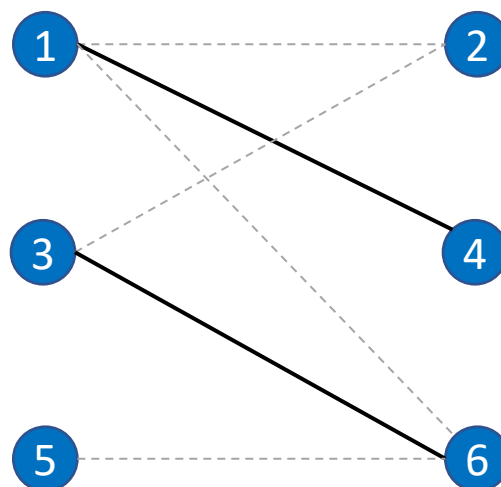
Path: 5,6,3,2



Augmenting Paths

- An **augmenting path** in a matching M is an alternating path **starting and ending in unmatched (free) vertices**

Path: 2,3,6,5



Augmenting Paths Property

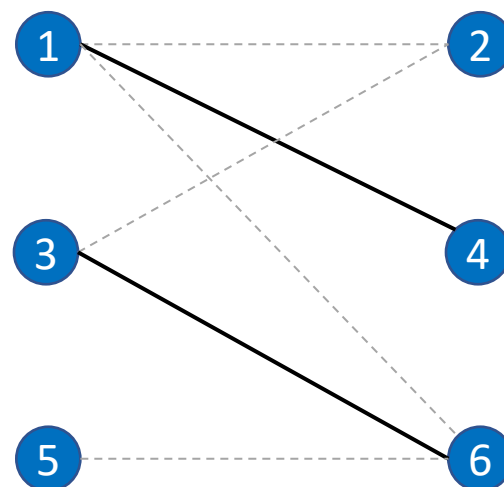
- In an augmenting path there is always one more non-matching edge than matching edges:

Path: 2,3,6,5

(2,3) unmatched

(3,6) **matched**

(6,5) unmatched



Augmenting Path Property (Contd.)

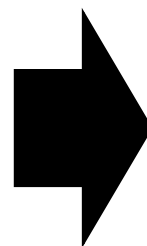
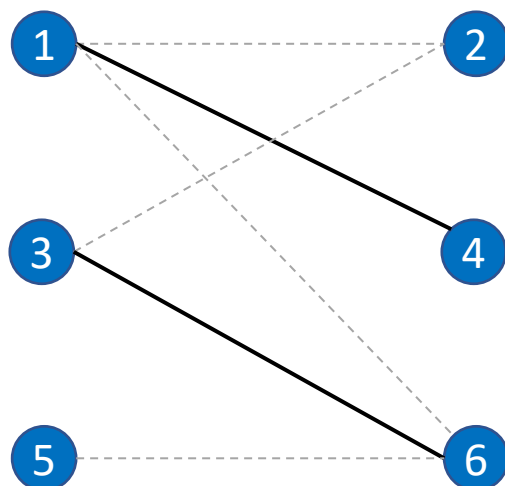
- We can use **augmenting paths** to **find maximum matchings**: just remove the matched edges from M and add the unmatched ones

Path: 2,3,6,5

(2,3) unmatched

(3,6) matched

(6,5) unmatched

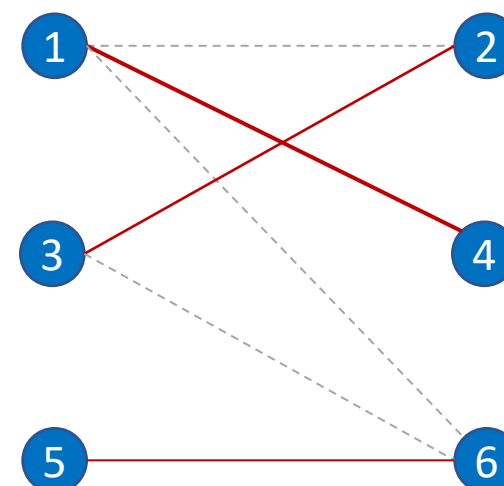


Path: 2,3,6,5

(2,3) unmatched

(3,6) matched

(6,5) unmatched



**We have found our
Maximum Matching!!!**

Augmenting Path Property (Contd.)

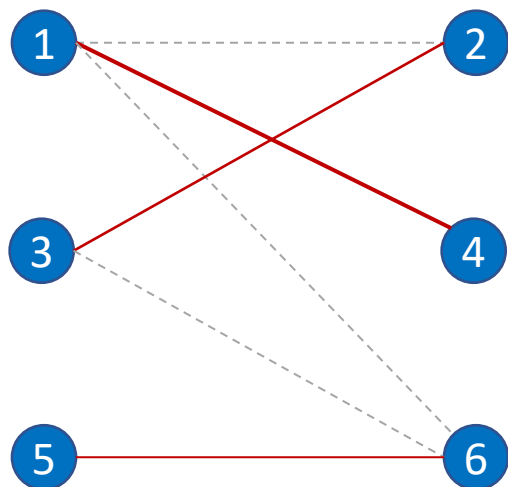
- If there is no *augmenting path*, then M is a *maximum matching*

Path: 2,3,6,5

(2,3) unmatched

(3,6) matched

(6,5) unmatched



We have found our Maximum Matching!!!

SUMMARY

- Matching Problem
- Maximal and Maximum Matching
- Alternating Paths
- Augmenting Paths

