Matchings

- A **matching** in an undirected graph is a set of vertex-disjoint edges.
- The **size** of a matching is the number of edges in it.
- A **maximum matching** of $G$ is a matching of maximum size.
A Committee of Committees

- The US senate has 20 committees and each senator may serve on several committees.
- The *committee of committees* should have a representative from each committee, and no senator is allowed to represent more than one committee.
- Is this always possible?
  - No! What if a senator is the only person on two committees?

A Committee of Committees?

- How can we find out whether a committee of committees is possible?
  - Build a graph!
A Committee of Committees?

- A committee of committees is possible if the graph has a matching of size 20.

Problem: Retreat Resort

- Problem. A retreat resort has $n$ guests staying in it. The resort offers hikes with travelling guides.
  - Every guest has a list of hikes that s/he is interested in.
  - Every guide is allowed to take up to 5 people.
  - Describe an efficient algorithm that finds whether every guest can go on a hike that s/he is interested in.
Building a Graph

• Create a bipartite graph with a vertex for every guest and for every hike.
  ◦ An edge between every guest and every hike that he is interested in.

Fixing the Graph

• A matching in the graph does not take into account that up to 5 people can go on a hike.
• Split every hike vertex $v$ into five vertices, and connect each of them to each of the vertices that $v$ was connected to.
• There is a valid hiking assignment if and only if the graph has a matching of size $n$. 
Reminder: Perfect Matchings

A perfect matching of a graph $G = (V, E)$ is a matching of size $|V|/2$.

Reminder: Neighbor Sets

Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. For any subset $A \subset V_1$, we define

$$N(A) = \{y \in Y \mid (x, y) \in E \text{ for some } x \in A\}.$$

$N(\{b, c, d\}) = \{u, v, w\}$

$N(\{a, e\}) = \{u, w, x\}$
Reminder: Variant of Hall's Theorem

- **Theorem.** Let $G = (V_1 \cup V_2, E)$ be a bipartite graph.
- There exists a matching of size $|V_1|$ in $G$ if and only if for every $A \subset V_1$, we have $|A| \leq |N(A)|$.

*Philip Hall*

Deficiency

- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph.
- The **deficiency** of $G$ is
  \[
  \text{def}(G) = \max_{A \subset V_1} \{|A| - |N(A)|\}.
  \]
- What is the deficiency of

\[
\begin{align*}
A &= \{b, c, d\} \\
\end{align*}
\]
Deficiency

- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph.
- The **deficiency** of $G$ is
  \[ \text{def}(G) = \max_{A \subseteq V_1} \{|A| - |N(A)|\}. \]
- The deficiency cannot be smaller than 0 since when $A = \emptyset$ we have
  \[ |A| - |N(A)| = 0. \]

Deficiency and Maximum Matchings

- **Theorem.** Let $G = (V_1 \cup V_2, E)$ be a bipartite graph. The **size of the maximum matching** in $G$ is
  \[ |V_1| - \text{def}(G). \]
- This implies **Hall’s theorem.**
  - $\text{def}(G) = 0$ if and only if there exists a matching of size $|V_1|$.
  - When $|V_1| = |V_2|$, we have $\text{def}(G) = 0$ if and only if there exists a perfect matching.
Proof: One Direction

• Set \( d = \text{def}(G) \).
• There exists a subset \( A \subset V_1 \) such that \( |A| - |N(A)| = d \).
• In any matching of \( G \), at least \( d \) vertices of \( A \) are unmatched.
• No matching can have size larger than \( |V_1| - d \).

It remains to prove that a matching of this size does exist.
Proof: The Other Direction

- We add \( d \) new vertices to \( V_2 \).
  - We connect every new vertex to each vertex of \( V_1 \).
- Originally, every set \( A \subset V_1 \) satisfied\[ |N(A)| \geq |A| - d.\]
  - Now \( |N(A)| \geq |A| \).
- By the variant of Hall’s theorem, there exists a matching \( M \) of size \( |V_1| \).
- Removing the new vertices, we obtain a matching of \( G \) of size \( |V_1| - d \).
The Size of a Maximum Matching

- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph.
- Can we use the deficiency theorem to find the size of the maximum matching of $M$?

- We can check $|A| - |N(A)|$ of every subset $A \subset V_1$.
  - But there are $2^{|V_1|}$ such subsets!

Alternating Paths

- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph, and let $M$ be a matching of $G$.
- A path is alternating for $M$ if it starts with an unmatched vertex of $V_1$ and every other edge of it is in $M$. 
Augmenting Paths

- An alternating path is **augmenting** for \( M \) if it also ends with an unmatched vertex.
- In an augmenting path, by switching the edges that are in \( M \) with the edges that are not, we obtain a **larger matching**.

Existence of Augmenting Paths

- **Theorem.** If a matching \( M \) in a bipartite graph \( G = (V_1 \cup V_2, E) \) is **not a maximum matching**, then there exists an **augmenting path** for \( M \).
- **Proof.**
  - Let \( M^* \) be a maximum matching of \( G \).
  - Let \( F \) be the set of edges that are either in \( M \) or in \( M^* \), but **not in both**.
  - In the graph \( G' = (V, F) \), every vertex is of degree at most two.
Example: The Graph $G'$

- The graph $G' = (V, F)$.
  - Every vertex has a degree of at most two.
  - The graph is a union of disjoint paths and cycles.

Finding an Augmenting Path

- By definition, $M^*$ has more edges than $M$.
- In at least one of the paths of $G'$, $M^*$ has more edges than $M$.
- This is an augmenting path for $M$!
Find a Maximum Matching

- We rely on the **theorem** to obtain an algorithm for finding a maximum matching.
- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph.
- Start with any matching $M$. A single edge is fine.
- Repeatedly find an **augmenting path** for $M$ and use it to obtain a larger matching.
- The process terminates after **at most** $|V_1|$ steps.

Finding an Augmenting Path

- Let $G = (V_1 \cup V_2, E)$ be a bipartite graph, and let $M$ be a matching.
- We wish to find whether there is an augmenting path for $M$ **starting at a specific unmatched vertex** $p_0 \in V_1$.
  - We run a variant of BFS from $p_0$. 

BFS Variant

- The root of the BFS tree is $p_0$.
- At the first level we have vertices that are adjacent to $p_0$ in $G$.

BFS Variant (2)

- For each vertex of level 1, if it is matched in $M$, we connect it to its match.
BFS Variant (3)

- For each vertex of level 2, we connect it (by edges not in $M$) to any of its neighbors in $G$ that are not yet in the tree.

BFS Variant (4)

- We repeat this process:
  - Vertices of **even levels** ($p_i$’s) have as their children every new vertex adjacent to them.
  - Vertices of **odd levels** ($q_i$’s) have only their matching vertex as a child.
BFS Variant (5)

• How can we tell whether an augmenting path for $M$ starts at $p_0$?
  ◦ Every such path corresponds to an unmatched vertex at an odd level of the tree (a leaf at an odd level).

Concluding Remarks

• Given a matching $M$ in a bipartite graph $G = (V_1 \cup V_2, E)$, for every vertex of $V_1$ that is unmatched in $M$:
  ◦ Run the BFS variant to check whether there is an augmenting path starting from it.

• If no augmenting paths were found — $M$ is a maximum matching.

• Otherwise, we use an augmenting path to obtain a larger matching and repeat.
The End

Goodbye cruel world