Chap 5 Matching and Factors

Yih-Lang Li (李毅郎)
Computer Science Department
National Yang-Ming Chiao-Tung University, Taiwan

The sources of most figure images are from the textbook

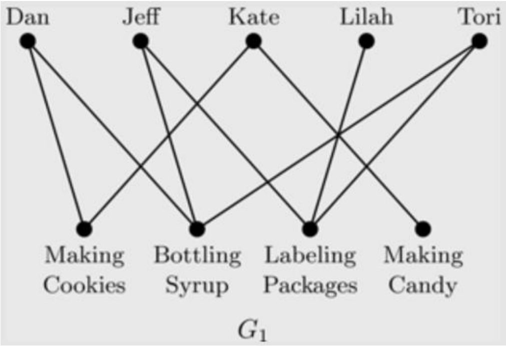
Outline

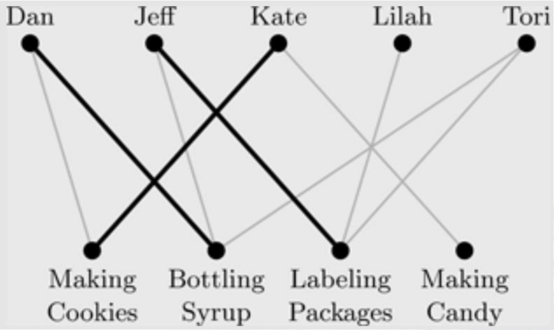
- Matching in Bipartite Graphs
- Matching in General Graphs
- Stable Matching
- Factors

5.1 Matching in Bipartite Graphs

- **Definition 5.1** Given a graph $G=(V, E)$, a *matching* M is a subset of the edges of G so that no two edges share an endpoint. The size of a matching, denoted $|M|$, is the number of edges in the matching.
  - ✓ A matching is just a set of independent edges within a graph.
- Example 5.2 The Vermont Maple Factory just received a rush order for 6-dozen boxes of maple cookies, 3-dozen bags of maple candy, and 10-dozen bottles of maple syrup. Some employees have volunteered to stay late tonight to help finish the orders. In the chart below, each employee is shown along with the jobs for which he or she is qualified. Draw a graph to model this situation and find a matching.

Employee	Task	
Dan	Making Cookies	Bottling Syrup
Jeff	Labeling Packages	Bottling Syrup
Kate	Making Candy	Making Cookies
Lilah	Labeling Packages	
Tori	Labeling Packages	Bottling Syrup

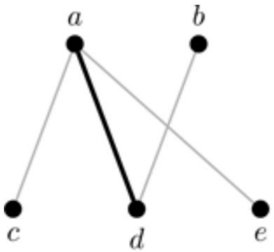




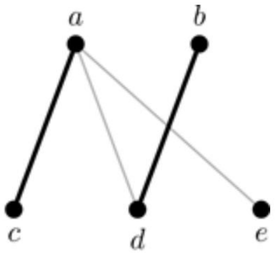
- **Definition 5.2** A vertex is *saturated* by a matching M if it is incident to an edge of the matching; otherwise, it is called *unsaturated*.

Matching in Bipartite Graphs

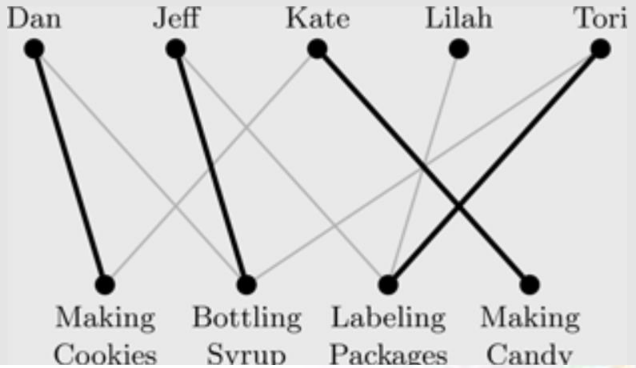
- **Definition 5.3** Given a matching M on a graph G, we say M is
  - ✓ *maximal* if M cannot be enlarged by adding an edge.
  - ✓ *maximum* if M is of the largest size amongst all possible matchings.
  - ✓ *perfect* if M saturates every vertex of G.
  - ✓ an *X-matching* if it saturates every vertex from the collection of vertices X (a similar definition holds for a Y-matching).
  - ✓ a perfect matching is automatically maximum and a maximum matching is automatically maximal, though the reverse need not be true.



Maximal Matching of G_2



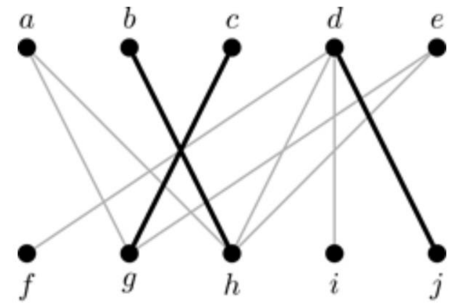
Maximum Matching of G_2



- **Example 5.2** Determine and find the proper type of matching for the Vermont Maple Factory graph G_1 from Example 5.1.
  - ✓ Every task must be assigned to an employee (X-matching for tasks)

Matching in Bipartite Graphs

- Is there a larger matching based on current state?
  - ✓ Given a set of vertices S, the neighbor set $N(S)$ consists of all the vertices incident to at least one vertex from S. For the graph G_3 above, if we consider $S=\{f, i, j\}$, then $N(S)=\{d\}$.
  - ✓ Maximum matching is at most 3.



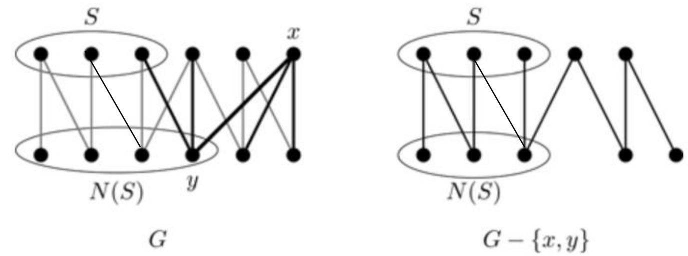
G_3

- **Theorem 5.4** (*Hall's Marriage Theorem*) Given a bipartite graph $G=(X \cup Y, E)$, there exists an X-matching if and only if $|S| \leq |N(S)|$ for any $S \subseteq X$.
  - ✓ X-matching $\rightarrow |S| \leq |N(S)|$
  - ✓ $|S| \leq |N(S)| \rightarrow$ Prove by induction on $|X|$, $|X| = 1$ it is trivially true. Now consider X.
  - ✓ Since $|S| \leq |N(S)|$, we discuss it in two cases: $|S| \leq |N(S)| - 1$ and $|S| = |N(S)|$
  - ✓ Case 1: $|S| \leq |N(S)| - 1$, $|N(S)| \geq |S| + 1$, thus for $x \in X$, there must exist y in Y such that $xy \in E$. Let $G' = G - \{x, y\}$

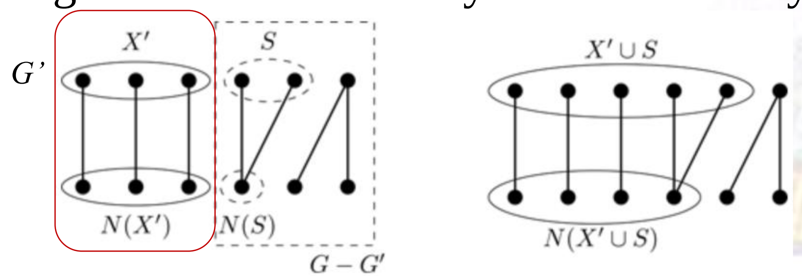
Matching in Bipartite Graphs

- ✓ Consider any $S \subseteq X-\{x\} \to |N_{G'}(S)| \geq |N_G(S)| - 1 \geq |S|$, and so G' satisfies the marriage condition.
- ✓ By IH,$\exists$ X-matching M' in G'

  $\to$ X-matching M in G



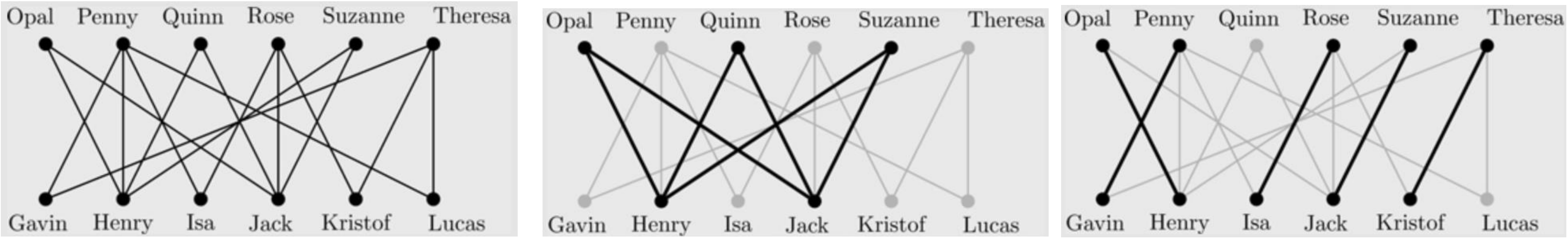
- ✓ Case 2: $|N(X')|=|X'|$ for any non-empty proper subset X' of X. Let G' be the subgraph as follows, then G' satisfies the marriage condition and by IH, G' has an X-matching. It remains to show that $G-G'$ also satisfies the marriage condition.
  - ➢ Another way to prove: claim: for two disjoint X_1 and X_2 sets, $N(X_1)$ and $N(X_2)$ are also disjoint,
- ✓ Let $S \subseteq X-X'$. Assume $|S|>|N_{G-G'}(S)|$, $|S \cup X'| = |S|+|X'|$ and $|N_G(S \cup X')| =|N(X')| + |N_{G-G'}(S)|< |N(X')|+|S|$. Thus $|S \cup X'|>|N_G(S \cup X')|$, which contradicts that G satisfies the marriage condition.
- ✓ Thus $G-G'$ also satisfies the marriage condition and so by the induction hypothesis has a matching that saturates $X-X'$.



Matching in Bipartite Graphs

- **Example 5.3**

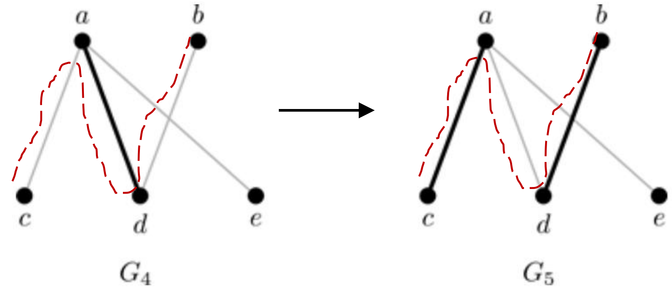
Girls	Boys She Likes			
Opal	Henry	Jack		
Penny	Gavin	Isa	Henry	Lucas
Quinn	Henry	Jack		
Rose	Kristof	Isa	Jack	
Suzanne	Henry	Jack		
Theresa	Gavin	Lucas	Kristof	

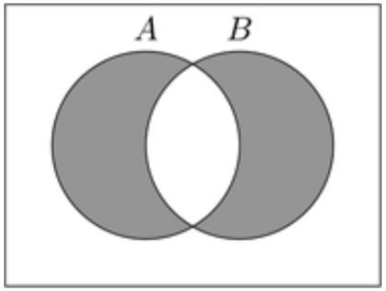


- **Corollary 5.5** Every k-regular bipartite graph has a perfect matching for all $k>0$.

Augmenting Paths and Vertex Covers

- **Definition 5.6** Given a matching M of a graph G, a path is called
  - ✓ *M-alternating* if the edges in the path alternate between edges that are part of M and edges that are not part of M.
  - ✓ *M-augmenting* if it is an M-alternating path and both endpoints of the path are unsaturated by M, implying both the starting and ending edges of the path are not part of M.

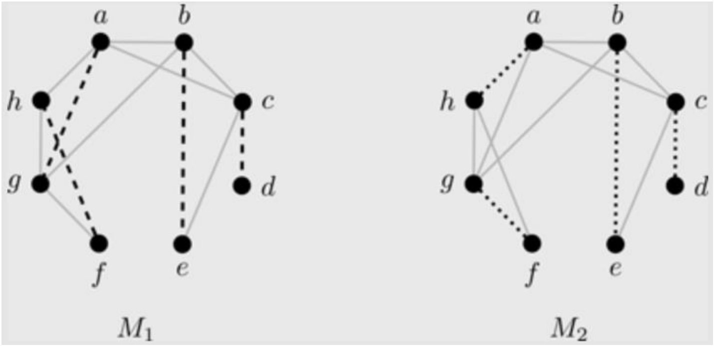


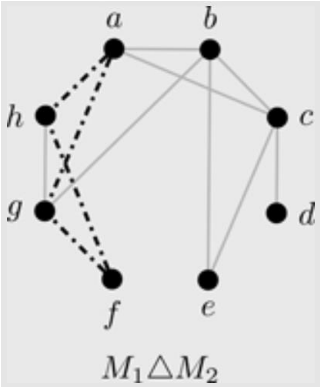


- **Theorem 5.7** (*Berge's Theorem*) A matching M of a graph G is maximum if and only if G does not contain any M-augmenting paths.
- **Definition 5.8** Let A and B be two sets. Then the symmetric difference $A \triangle B$ is all those elements in exactly one of A and B; that is, $A \triangle B = (A - B) \cup (B - A)$.

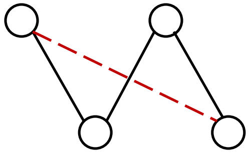
Augmenting Paths and Vertex Covers

- **Example 5.4** Below are two different matchings of a graph G. Find $M_1 \triangle M_2$.





- **Lemma 5.9** Let M_1 and M_2 be two matchings in a graph G. Then every component of $M_1 \triangle M_2$ is either a path or an even cycle.
  - ✓ Let $H = M_1 \triangle M_2$. $deg(v) \leq 2$ for every vertex v in H.
  - ✓ Thus every component of H consists of paths and cycles.
  - ✓ Every cycle must be of even length.



Augmenting Paths and Vertex Covers

- **Theorem 5.7** (*Berge's Theorem*, restated) A matching M of a graph G is not maximum if and only if G contains some M-augmenting path.
  - $\leftarrow$, suppose M is a matching of G and G contains an M-augmenting path $P \rightarrow$ a larger matching M' and so M is not a maximum matching of G.
  - $\rightarrow$, suppose M is a matching of G and M is not maximum, $\exists\ M'$, $|M'| > |M|$. We can make an M-augmenting path by looking at $M \triangle M'$.
  - By Lemma 5.9, every component of $M \triangle M'$ is either a path or even cycle. Since $|M'|>|M|$, $\exists$ a component C that is path starting and ending at M' but not an even cycle, implying the component C is an M-augmenting path.

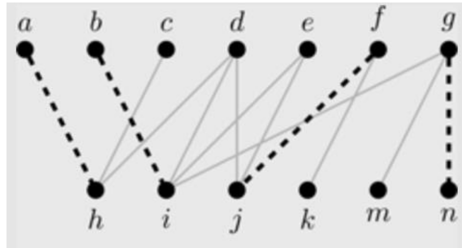
Augmenting Paths and Vertex Covers

Augmenting Path Algorithm

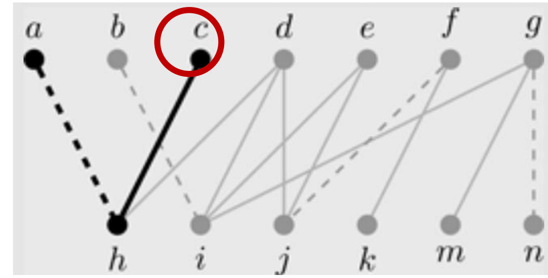
- *Input*: Bipartite graph $G=(X \cup Y, E)$.
- *Steps*:
  1. Find an arbitrary matching M.
  2. Let U denote the set of unsaturated vertices in X.
  3. If U is empty, then M is a maximum matching; otherwise, select a vertex x from U.
  4. Consider y in $N(x)$.
  5. If y is also unsaturated by M, then add the edge xy to M to obtain a larger matching M'. Return to Step (2) and recompute U. Otherwise, go to Step (6).'
  6. If y is saturated by M, then find a maximal M-alternating path from x using xy as the first edge.
     (a) If this path is M-augmenting, then switch edges along that path to obtain a larger matching M'; that is, remove from M the matched edges along the path and add the unmatched edges to create M'. Return to Step (2) and recompute U.
     (b) If the path is not M-augmenting, return to Step (4), choosing a new vertex from $N(x)$.
  7. Stop repeating Steps (2)–(4) when all vertices from U have been considered.
- *Output*: Maximum matching for G.

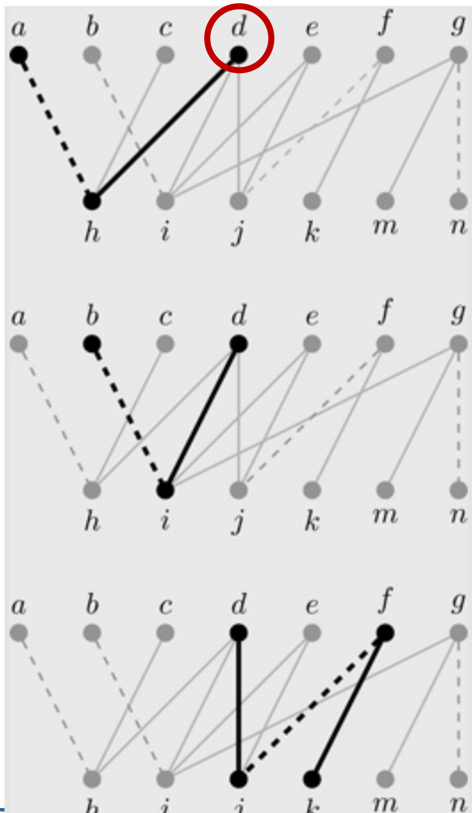
Augmenting Paths and Vertex Covers

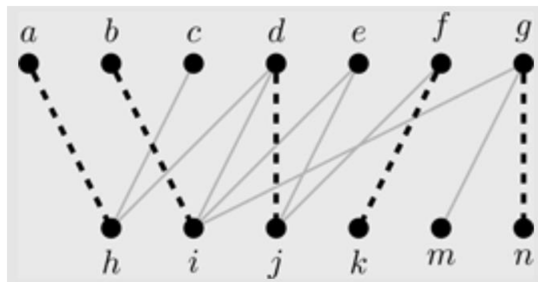
- **Example 5.5** Apply the Augmenting Path Algorithm to the bipartite graph G_6 below

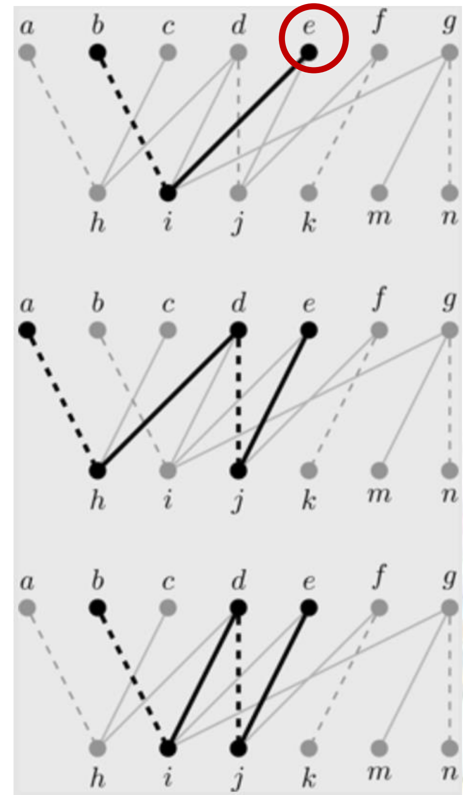


G_6



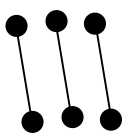




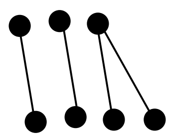


Augmenting Paths and Vertex Covers

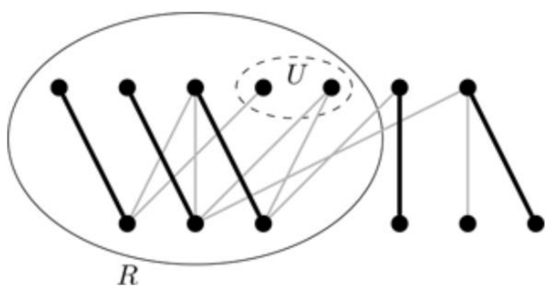
- **Definition 5.10** A vertex cover Q for a graph G is a subset of vertices so that every edge of G has at least one endpoint in Q.
- **Theorem 5.11** (*König-Egerváry Theorem*) For a bipartite graph G, the size of a maximum matching of G equals the size of a minimum vertex cover for G.
  - ✓ Let $G=(X\cup Y, E)$ be a bipartite graph with maximum matching M and minimum vertex cover Q. $|Q|\geq|M|$. Find a vertex cover Q' such that $|Q'|\leq |M| \rightarrow |Q|=|M|$.
  - ✓ U: the set of unsaturated vertices in X, R: the set of vertices that can be reached by an M-alternating path that begins at a vertex in U. $U\subseteq R$ and any vertex in $X\cap R-U$ must be saturated.

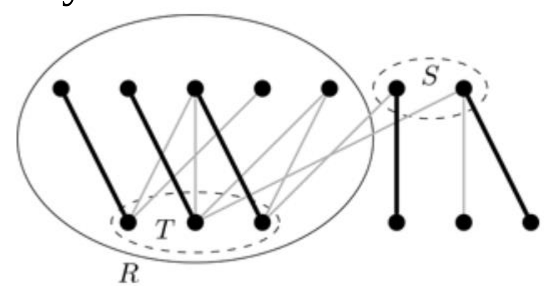


$|Q|=|M|$



$|Q|\geq|M|$

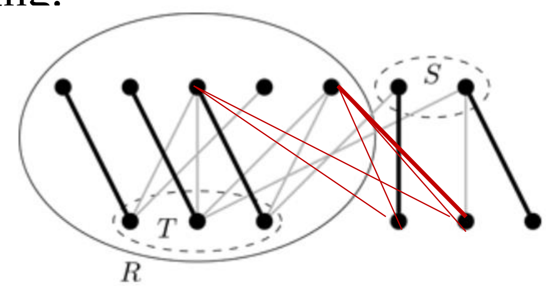




  - ✓ Define $S=X-R$ and $T=Y\cap R$. We will prove $S\cup T$ is a Q' and $|Q'|\leq |M|$.
  - ✓ Let $v\in S\cup T$, $\forall v\in T$, v is saturated, otherwise an edge from U to v would enlarge M. $\forall v\in S$, v is saturated, otherwise $v\in U$ but not S. $\forall v\in S\cup T$, v is saturated $\rightarrow |S\cup T|\leq|M|$.

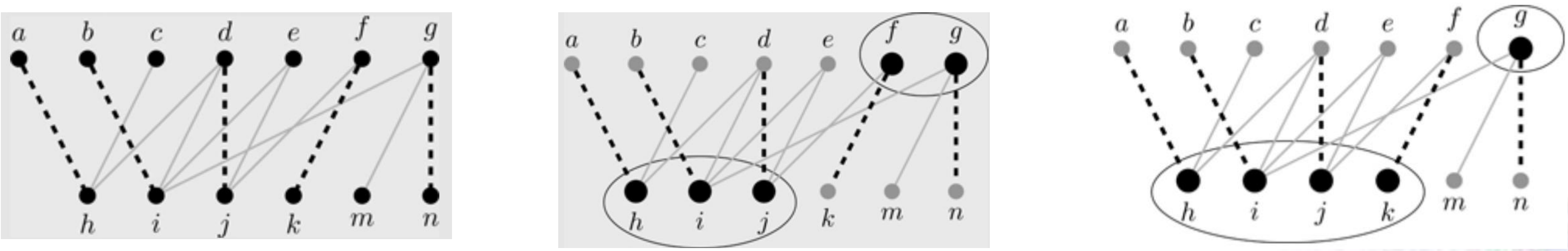
Augmenting Paths and Vertex Covers

- Consider any edge $e=xy$ of G where $x \in X$ and $y \in Y$, if either $x \in S$ or $y \in T$, $S \cup T$ is a vertex cover. If $x \in S$ then we are done. Otherwise $x \in R \rightarrow y \in T$ since $X \cap R$ do not have edges linking with Y-T (can skip the following discussion).
- If $e \in M$ ($x \in R$) then we know both x and y are saturated. Thus $x \notin U$, but there exists some M-alternating path P from a vertex in U to x. This path must use edge e and so y is also part of P, and so y is also reachable from U by an M-alternating path, making $y \in T$.
- Otherwise $e \notin M$ ($x \in R$). If $x \in U$ then e is itself an M-alternating path and so $y \in T$. Otherwise there exists an M-alternating path P from U to x, and either e is a part of P or the last edge of P must be an edge from M and $e=xy$ can be added to the end of this path. In either case $y \in T$ .Thus every edge of G has an endpoint in $S \cup T$ and so this is a vertex cover of size $|M|$, proving a minimum vertex cover must be of the same size of a maximum matching.



Augmenting Paths and Vertex Covers

- **Vertex Cover Method**
  - ✓ Let $G=(X \cup Y, E)$ be a bipartite graph.
  - ✓ Apply the Augmenting Path Algorithm and mark the vertices considered throughout its final implementation.
  - ✓ Define a vertex cover Q as the unmarked vertices from X and the marked vertices from Y.
  - ✓ Q is a minimum vertex cover for G.
- **Example 5.6** Apply the Vertex Cover Method to the output graph from Example 5.5.

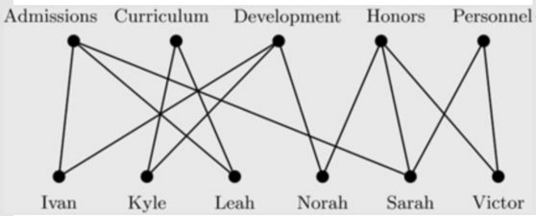


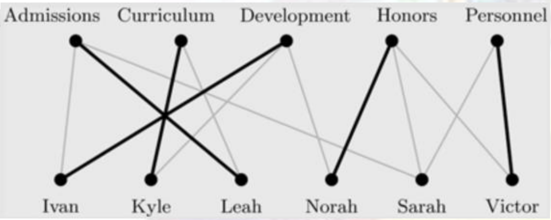
  - ✓ The marked vertices from X are a,b,c,d, and e, and the marked vertices from Y are h,i, and j.

Augmenting Paths and Vertex Covers

- **Definition 5.12** Given a collection of finite nonempty sets $S_1,S_2,\ldots,S_n$ (where $n\geq 1$), a system of distinct representatives, or *SDR*, is a collection $r_1,r_2,\ldots,r_n$ so that r_i is a member of set S_i and $r_i\neq r_j$ for all $i\neq j$ (for all $i,j=1,2,\ldots,n$).
- **Example 5.7** Find an *SDR* for the collection of sets given below: $S_1=\{1,2,3,5\}$, $S_2=\{2,4,8\}$, $S_3=\{2,6\}$, $S_4=\{4,8\}$.
  - ✓ One possible system of distinct representatives is: $r_1=1,\ r_2=2,\ r_3=6,\ r_4=4$. Note that another solution is: $r_1=1,\ r_2=4,\ r_3=2,\ r_4=8$.
- **Theorem 5.13** A collection of finite nonempty sets $S_1,S_2,\ldots,S_n$ (where $n\geq 1$) has an *SDR* if and only if $|\bigcup_{i\in R} S_i|\geq|R|$ for all $R\subseteq\{1, 2,\ldots, n\}$.
- **Example 5.8** During faculty meetings at a small liberal arts college, multiple committees provide a report to the faculty at large. These committees often overlap in membership, so it is important that, for any given year, a person is not providing the report for more than one committee. Find a system of distinct representatives for the groups listed below.

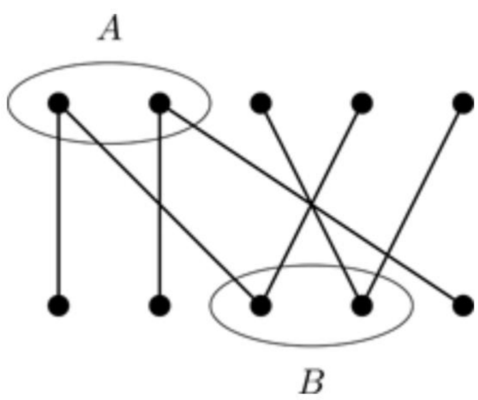
Committee	Members		
Admissions Council	Ivan	Leah	Sarah
Curriculum Committee	Kyle	Leah	
Development and Grants	Ivan	Kyle	Norah
Honors Program Council	Norah	Sarah	Victor
Personnel Committee	Sarah	Victor	

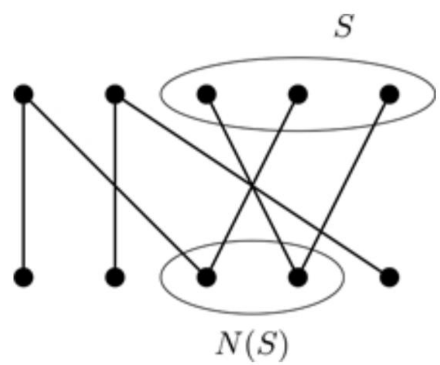




Hall's Theorem Revisited

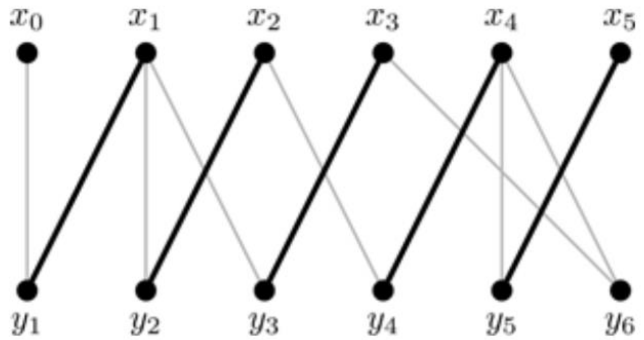
- **Theorem 5.4** (*Hall's Marriage Theorem*) Given a bipartite graph $G=(X \cup Y, E)$, there exists an X-matching if and only if $|S| \leq |N(S)|$ for any $S \subseteq X$.
  - ✓ $\leftarrow$ prove by contrapositive: If G does not have an X-matching then there exists some $S \subseteq X$ such that $|S|>|N(S)|$.
  - ✓ Assume G does not have an X-matching. By Theorem 5.11, $\exists$ a vertex cover Q with $|Q|<|X|$.
  - ✓ Let $Q=A \cup B$ where $A \subseteq X$ and $B \subseteq Y$. Then $|A|+|B|=|Q|<|X|$ and so $|B|<|X|-|A|=|X-A|$
  - ✓ $\nexists e$ to connect vertices in $X-A$ and $Y-B$.
  - ✓ Let $S=X-A$. $\forall\ v \in$ S , $N(v) \subseteq$ B. Thus $|N(S)| \leq |B| < |X-A| = |S|$.





Hall's Theorem Revisited

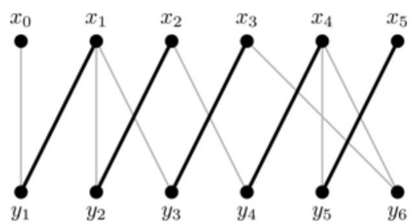
- We will construct a sequence of distinct vertices $x_0, y_1, x_1, y_2, \ldots$, with $x_i \in X$ and $y_i \in Y$ so that the following three conditions are met:
  - ✓ x_0 is unsaturated.
  - ✓ for all $i \geq 1$, y_i is adjacent to some vertex $x_{f(i)} \in \{x_0, \ldots, x_{i-1}\}$.
  - ✓ $x_i y_i \in M$ for all $i \geq 1$.
  - ✓ $y_6 x_{f(6)}$ $(f(6) = 4)$. $x_4 \sim y_4 \ \rightarrow \ y_6 x_{f(6)} y_{f(6)}$
  - ✓ $x_2 \sim y_4$ and $y_2 \sim x_2$, $path\ y_6 x_4 y_4 x_2 y_2 \rightarrow y_6 x_{f(6)} y_{f(6)} x_{f(f(6))} y_{f(f(6))}$
  - ✓ $y_2 \sim x_1$, $x_1 \sim y_1$, $and\ y_1 \sim x_0$, $path\ y_6 x_4 y_4 x_2 y_2 x_1 y_1 x_0 \rightarrow$
    $y_6 x_{f(6)} y_{f(6)} x_{f(f(6))} y_{f(f(6))} x_{f(f(f(6)))} y_{f(f(f(6)))} x_{f(f(f(f(6))))}$
  - ✓ Or simply $y_6 x_{f(6)} y_{f(6)} x_{f^2(6)} y_{f^2(6)} x_{f^3(6)} y_{f^3(6)} x_{f^4(6)}$ ($f^4(6) = 0$)



Hall's Theorem Revisited

Hall's Theorem

- Assume $|N(S)|\geq|S|$ for all $S\subseteq X$. Suppose for a contradiction that M is a maximum matching that leaves a vertex of X unsaturated.
- Let $x_0, y_1, x_1, y_2,\ldots$, with $x_i\in X$ and $y_i\in Y$ be a maximal sequence of distinct vertices so that the following three conditions are met:
  - x_0 is unsaturated.
  - for all $i\geq 1$, y_i is adjacent to some vertex $x_{f(i)}\in\{x_0,\ldots,x_{i-1}\}$.
  - $x_iy_i\in M$ for all $i\geq 1$.

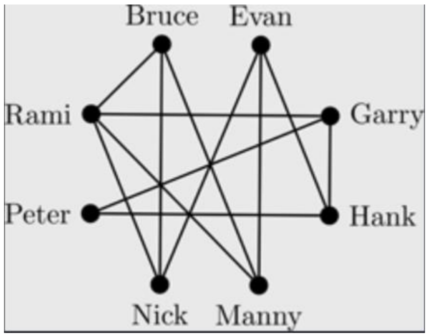


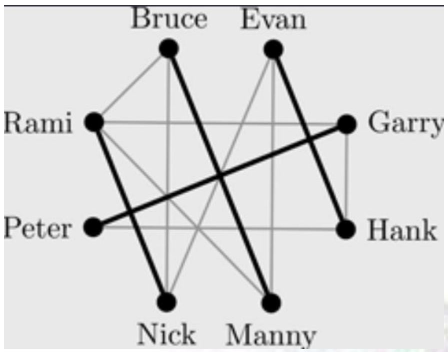
- Let z be the final vertex in our sequence. If $z=x_k$ then $S=\{x_0, x_1,\ldots, x_k\}$ has size $k+1$, and by $|N(S)|\geq k+1$, $\exists y\notin\{y_1, \ldots, y_k\}$ that is adjacent to some x_i from S. Thus $y_{k+1}= y$, contradicting the maximality of the length of the sequence.
- Thus the final vertex z of the sequence is in Y, call it y_k. Form the path $P=y_kx_{f(k)}y_{f(k)}x_{f^2(k)}\ldots y_{f^{r-1}(k)}x_{f^r(k)}$, $f^r(k)=0$. Edges $(y_{f^{r-1}(k)}x_{f^r(k)})$ and $(y_kx_{f(k)})$ are unmatched. y_k is unsaturated $\rightarrow$ P is an augmenting path.
- If y_k were saturated, $\exists x\in X$ such that $y_kx\in M$. Since y_k is the last vertex of the sequence, $x\in\{x_0, x_1, \ldots, x_{k-1}\}$. $y_kx_i\in M$ and $i\neq k$, but only $y_kx_k\in M$, contradiction. Thus y_k and x_0 are both unsaturated, making P an augmenting path.
- This contradicts Berge's Theorem since M was chosen to be a maximum matching. Thus every vertex in X must be saturated and so M is an X-matching.

5.2 Matching in General Graphs

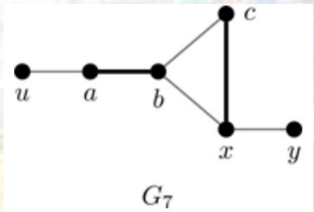
- Bruce, Evan, Garry, Hank, Manny, Nick, Peter, and Rami decide to go on a week-long canoe trip in Guatemala. They must divide themselves into pairs, one pair for each of four canoes, where everyone is only willing to share a canoe with a few of the other travelers.
- Example 5.9 The group of eight men from above have listed who they are willing to share a canoe with. This information is shown in the following table, where a Y indicates a possible pair. Note that these relationships are symmetric, so if Bruce will share a canoe with Manny, then Manny is also willing to share a canoe with Bruce. Model this information as a graph. Find a perfect matching or explain why no such matching exists.

	Bruce	Evan	Garry	Hank	Manny	Nick	Peter	Rami
Bruce	·	·	·	·	Y	Y	·	Y
Evan	·	·	·	Y	Y	Y	·	·
Garry	·	·	·	Y	·	·	Y	Y
Hank	·	Y	Y	·	·	·	Y	·
Manny	Y	Y	·	·	·	·	·	Y
Nick	Y	Y	·	·	·	·	·	Y
Peter	·	·	Y	Y	·	·	·	·
Rami	Y	·	Y	·	Y	Y	·	·



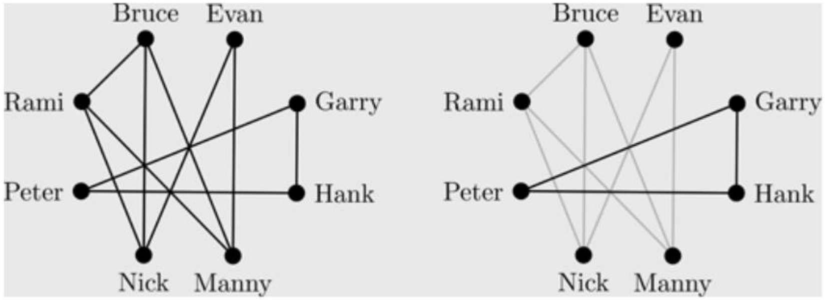


- Berge's Theorem still holds but alternative path finding becomes complex.
  - ✓ $u \rightarrow x$: *uabx* or *uabcx*. The latter is better if moving one more step as *uabcxy*



Matching in General Graphs

- **Example 5.10** Halfway through the canoe trip from Example 5.23, Rami will no longer share a canoe with Garry, and Hank angered Evan so they cannot share a canoe. Update the graph model and determine if it is now possible to pair the eight men into four canoes.



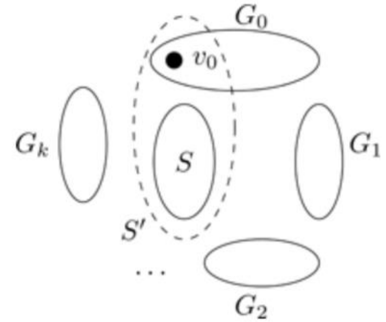
- **Definition 5.14** Let G be a graph. Define $o(G)$ to be the number of odd components of G, that is the number of components containing an odd number of vertices.
  - ✓ In Example 5.10, there are two odd components.
- **Theorem 5.15** (*Tutte's Theorem*) A graph $G=(V, E)$ has a perfect matching if and only if for every proper subset of vertices S the number of odd components of $G-S$ is at most $|S|$, that is $o(G-S)\leq|S|$.
  - ✓ $o(G-S)\leq|S|$ for every $S\subsetneq V(G)$.
  - ✓ $\rightarrow$ suppose G has a perfect matching and S is any subset of vertices from G. $\forall C_o$, $\exists v_1$, v_2, v_1 in C_o and v_2 in S such that (v_1, v_2) in E. Thus $o(G-S)\leq|S|$ for every $S\subsetneq V(G)$.

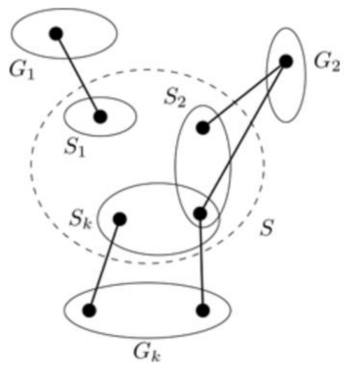
Matching in General Graphs

- $\leftarrow$ assume $|G|=n$. We will show that G has a perfect matching whenever it satisfies Tutte's condition.
- if $S=\emptyset$, then $|S|=0$ and $G-S=G$. Thus $o(G)=o(G-S)\leq|S|=0 \rightarrow G$ only have even components. Thus n is even. Moreover, $|S|$ and $o(G-S)$ must have the same parity.
- Prove by induction on n (even integer) that if G satisfies Tutte's condition then G will have a perfect matching.
- First suppose $n=2$. no odd components $\rightarrow G=K_2$ and G has a perfect matching. Now suppose for some $n\geq 4$ that all graphs H of even order $n'<n$ satisfying Tutte's condition have a perfect matching. Let G be a graph of order n satisfying Tutte's condition. We will consider whether $o(G-S)<|S|$ for $\forall S\subsetneq V(G)$ or if $\exists S\subsetneq V(G)$ where $o(G-S)=|S|$. Note we only need to consider sets S with $2\leq S\leq n$.
- *Case* 1: Suppose $o(G-S)<|S|$ for all $S\subsetneq V(G)$.
  - $o(G-S)\leq|S|-2$ for all $S\subsetneq V(G)$ with $2\leq|S|\leq n$.
  - Let $e=xy$ in G and let $G'=G-\{x, y\}$. Let $T\subsetneq V(G')$ and let $T'=T\cup\{x, y\}$. Then $|T'|=|T|+2$.
  - If $o(G'-T)>|T|$ then $o(G'-T)>|T'|-2$. $o(G'-T)=o(G-T') \rightarrow o(G-T')\geq|T'|$, a contradiction.
  - Thus $o(G'-T)\leq|T|$, and so by the induction hypothesis G' has a perfect matching M. Together with the edge $e=xy$, $M\cup e$ is a perfect matching of G.

Matching in General Graphs

- *Case* 2: Suppose there exist some $S' \subsetneq V(G)$ satisfying (*) $o(G-S')=|S'|$. *Among* all possible sets S', pick S to be the largest possible one satisfying condition (*), say $|S|=k$.
- G-S only consists of odd components.
  - Let $G_1, G_2, \ldots, G_k$ be the odd components of $G-S$. Suppose G_0 were some even component of $G-S$, and let $v_0 \in V(G_0)$. Then $o(G_0-\{v_0\}) \geq 1$.
  - Let $S'=S \cup \{v_0\}$. Thus $o(G-S') \geq k+1=|S'|$, and we can find one G_0 for $o(G-S')=|S'|$, S' is a larger set than S satisfying (*), a contradiction to our choice of S. Thus $G-S$ can only consist of odd components.

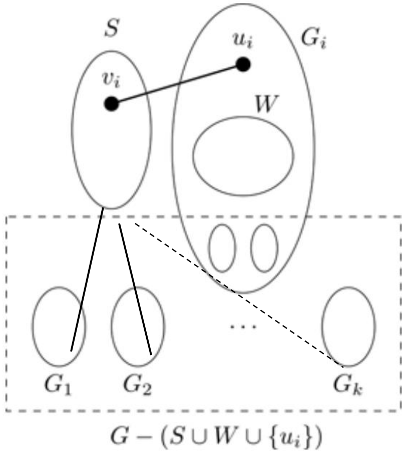




- There exists a system of distinct representatives for $S_1, \ldots, S_k$
  - For $i=1, \ldots, k$, let S_i denote the vertices of S adjacent to at least one vertex of G_i. Since $o(G)=0$, $S_i \neq \emptyset$ for all i.
  - let R be the union of j of the S_i's. Then $R \subseteq S$ and in $G-R$ the only odd components that remain are the G_i for which $S_i \subseteq R$. Thus $o(G-R)=j$ and since $o(G-R) \leq |R| \rightarrow |R| \geq j$.
  - By Theorem 5.13, $\exists$ *SDR* $(v_1, \ldots, v_k)$ for $S_1, \ldots, S_k$ such that $v_i \in S_i$ and for some $u_i \in G_i$ we have $v_i \sim u_i$.
  - We will show each $G_i - u_i$ has a perfect matching, which we can combine together with $v_i u_i$ to get a perfect matching of G.

Matching in General Graphs

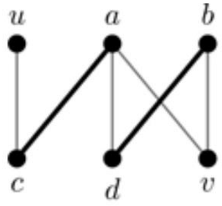
- ✓ $G_i{-}u_i$ satisfies Tutte's condition.
  - $G_i{-}u_i$ has an even number of vertices. For any i, consider $W \subsetneq V(G_i{-}u_i)$. If $o(G_i{-}u_i{-}W) > |W|$ then $o(G_i{-}u_i{-}W) \geq |W|+2$.
  - $G-(S \cup W \cup \{u_i\})$ has exactly the odd components of G-S (G_1 to G_k), except G_i, along with the odd components of $G_i{-}u_i{-}W \rightarrow o(G-(S \cup W \cup \{u_i\})) = o(G_i{-}u_i{-}W) + o(G-S) - 1 \geq |S|+|W|+1 = |S \cup W \cup \{u_i\}|$, contradicting the maximality of S.
  - Thus for each i we know $o(G_i{-}u_i{-}W) \leq |W|$, that is $G_i{-}u_i$ satisfies Tutte's condition. Thus by the induction hypothesis applied to every $G_i{-}u_i$ we know that $G_i{-}u_i$ has a perfect matching. Since $|S|=k$, v_1u_1, v_2u_2,…, v_ku_k forms a matching that saturates S and together with each of the perfect matchings of the $G_i{-}u_i$ we have a prefect matching of G.



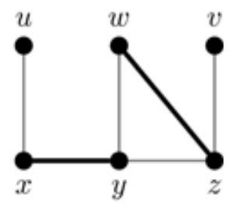
- **Corollary 5.16** Every cubic graph without any bridges has a perfect matching.

Edmonds' Blossom Algorithm

- The differences between bipartite graphs and simple graphs
  - ✓ All the alternating paths from u to b in G_8 will use the matching edge db and are of even length.
  - ✓ In G_9, the alternating paths from u to z can be of odd or even length ($uxyz$ and $uxywz$)

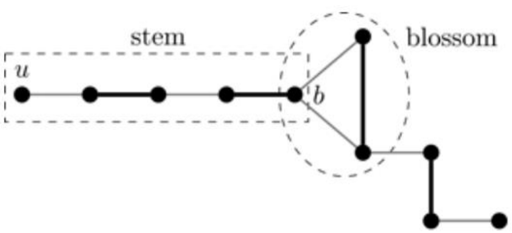


G_8



G_9

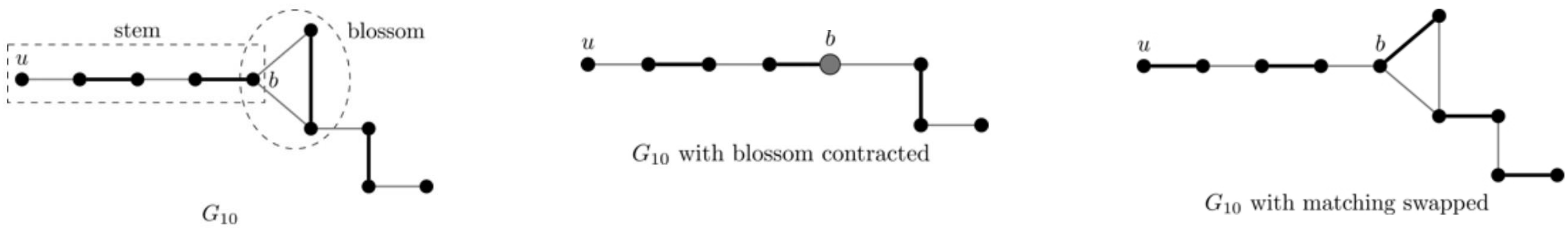
- **Definition 5.17** Given a graph G and a matching M, a *flower* is the union of two M-alternating paths from an unsaturated vertex u to another vertex v where one path has odd length and the other has even length. The *stem* of the flower is the maximal common initial path out of u, that ends at a vertex b, called the base. The *blossom* is the odd cycle that is obtained by removing the stem from the flower.
  - ✓ the stem must be of even length,
  - ✓ the blossom is an odd cycle C_{2k+1} with k edges from M,
  - ✓ every vertex of the blossom must be saturated by M,
  - ✓ Any non-stem edges coming off the blossom must be unmatched.



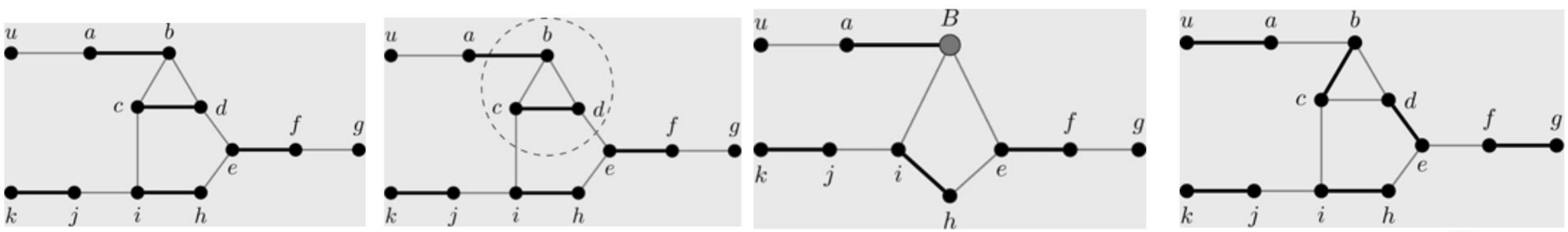
G_{10}

Edmonds' Blossom Algorithm

- Blossom contraction → augmenting path → edge swapping



- **Example 5.11** Use Edmonds' Blossom Algorithm to find a maximum matching on the graph below, where the initial matching is shown in bold.



uabc, uabd→uabdc
meet the last vertex of
previous path

Blossom contraction

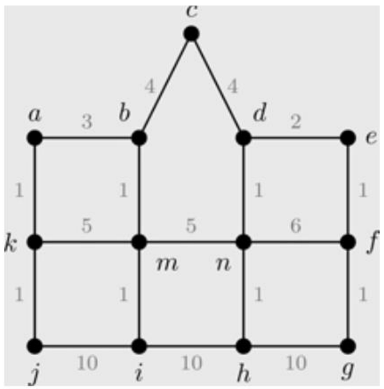
Augmenting path
Edge swapping

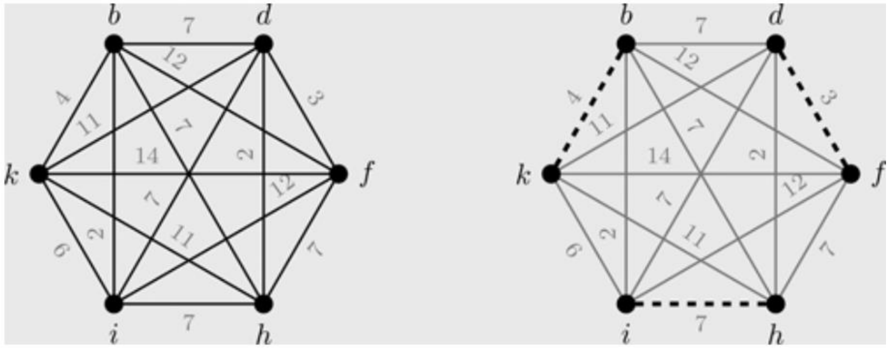
Chinese Postman Problem

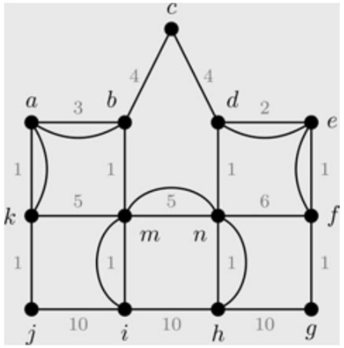
- Proposed by the Chinese mathematician Guan Meigu in 1960 and was solved about a decade later by Jack Edmonds and Ellis Johnson.
  - ✓ uses both Dijkstra's Algorithm for finding a shortest path (see Section 2.3.1) and a matching in a complete graph.
- **Postman Algorithm**
  - ✓ *Input*: Weighted graph $G=(V, E, w)$.
  - ✓ *Steps*:
    1. Find the set S of odd vertices in G.
    2. Form the complete graph K_n where $n=|S|$.
    3. For each distinct pair $x, y \in S$, find the shortest path P_{xy} and its total weight $w(P_{xy})$.
    4. Define the weight of the edge xy in K_n to be $w(xy)=w(P_{xy})$).
    5. Find a perfect matching M of K_n of least total weight.
    6. For each edge $e=xy \in M$, duplicate the edges of P_{xy} corresponding to the shortest path from x to y, creating G^*.
    7. Find an eulerian circuit of G^*.
  - ✓ *Output*: Optimal weighted-eulerization of G.

Chinese Postman Problem

- **Example 5.12** Apply the Postman Algorithm to the graph below.



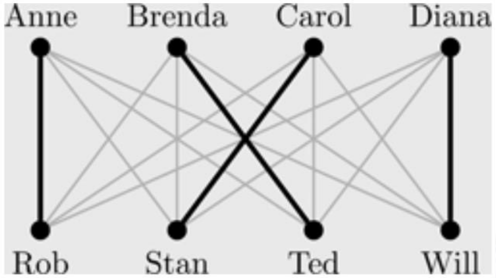


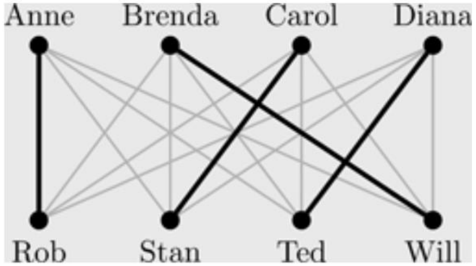


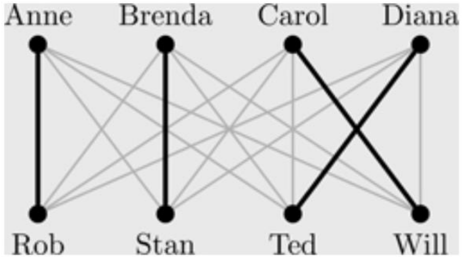
5.3 Stable Matching

- **Definition 5.18** A perfect matching is *stable* if no unmatched pair is unstable; that is, if x and y are not matched but both rank the other higher than their current partner, then x and y form an *unstable* pair.
  - ✓ In essence, when pairing couples into marriages we want to ensure no one will leave their current partner for someone else.
- Example 5.13 Four men and four women are being paired into marriages. Each person has ranked the members of the opposite sex as shown below. Draw a bipartite graph and highlight the matching Anne–Rob, Brenda–Ted, Carol–Stan, and Diana–Will. Determine if this matching is stable. If not, find a stable matching and explain why no unstable pair exists.

Anne:	$t > r > s > w$	Rob:	$a > b > c > d$
Brenda:	$s > w > r > t$	Stan:	$a > c > b > d$
Carol:	$w > r > s > t$	Ted:	$c > d > a > b$
Diana:	$r > s > t > w$	Will:	$c > b > a > d$







Stable Matching

Gale-Shapley Algorithm

- ✓ *Input*: Preference rankings of n women and n men.
- ✓ *Steps*:
  1. Each man proposes to the highest ranking woman on his list.
  2. If every woman receives only one proposal, this matching is stable. Otherwise move to Step (3).
  3. If a woman receives more than one proposal, she
     a. accepts if it is from the man she prefers above all other currently available men and rejects the rest; or,
     b. delays with a maybe to the highest ranked proposal and rejects the rest.
  4. Each man now proposes to the highest ranking unmatched woman on his list who has not rejected him.
  5. Repeat Steps (2)–(4) until all people have been paired.
- ✓ *Output*: Stable Matching.

Stable Matching

- **Example 5.14** Apply the Gale-Shapley Algorithm to the rankings from Example 5.13, which are reproduced below.

Anne:	t	>	r	>	s	>	w		Rob:	a	>	b	>	c	>	d
Brenda:	s	>	w	>	r	>	t		Stan:	a	>	c	>	b	>	d
Carol:	w	>	r	>	s	>	t		Ted:	c	>	d	>	a	>	b
Diana:	r	>	s	>	t	>	w		Will:	c	>	b	>	a	>	d

Rob	Anne	?
Stan	Anne	X
Ted	Carol	X
Will	Carol	V

Rob	Anne	V
Stan	Brenda	V
Ted	Diana	V
Will	Carol	V

- **Example 5.15** Apply the Gale-Shapley Algorithm to the rankings from Example 5.13 with the women proposing.

Anne	Ted	V
Brenda	Stan	V
Diana	Will	V
Carol	Rob	V

Unacceptable Partners

Anne:	t > r > s > w	Rob:	a > b > c > d
Brenda:	w > r > t	Stan:	a > b
Carol:	w > r > s > t	Ted:	c > d > a > b
Diana:	s > r > t	Will:	c > b > a

- **Gale-Shapley Algorithm (with Unacceptable Partners)**
  - ✓ *Input*: Preference rankings of *n* women and *n* men.
  - ✓ *Steps*:
    1. Each man proposes to the highest ranking woman on his list.
    2. If every woman receives only one proposal from someone they deem acceptable, they all accept and this matching is stable. Otherwise move to Step (3).
    3. If the proposals are not all different, then each woman:
       a. rejects a proposal if it is from an unacceptable man;
       b. accepts if the proposal is from the man she prefers above all other currently available men and rejects the rest; or
       c. delays with a maybe to the highest ranked proposal and rejects the rest.
    4. Each man now proposes to the highest ranking unmatched woman on their list who has not rejected him.
    5. Repeat Steps (2)–(4) until all people have been paired or until no unmatched man has any acceptable partners remaining.
  - ✓ *Output*: Stable Matching.

Unacceptable Partners

- **Example 5.16** Apply the Gale-Shapley Algorithm to the rankings on page 255 to find a stable matching.

Anne:	t > r > s > w	Rob:	a > b > c > d
Brenda:	w > r > t	Stan:	a > b
Carol:	w > r > s > t	Ted:	c > d > a > b
Diana:	s > r > t	Will:	c > b > a

  - ✓ Round 1: Rob → Anne, Stan → Anne, Ted → Carol, and Will → Carol.
  - ✓ Round 2: Rob → Anne, Stan → Brenda, Ted → Diana
  - ✓ Round 3: Rob → Anne, Stan → ?, Ted → Diana

Rob	Anne	?
Stan	Anne	X
Ted	Carol	X
Will	Carol	V

Rob	Anne	?
Stan	Brenda	X
Ted	Diana	?
Will	Carol	V

Rob	Anne	V
Stan		
Ted	Diana	V
Will	Carol	V

Stable Roomates

- **Example 5.17** Four women are to be paired as roommates. Each woman has ranked the other three as shown below. Find all possible pairings and determine if any are stable.

Emma:	l	>	m	>	z
Leena:	m	>	e	>	z
Maggie:	e	>	z	>	l
Zara:	e	>	l	>	m

  - ✓ Emma ↔ Leena and Maggie ↔ Zara: This is stable since Emma is with her first choice and the only person Leena prefers over Emma is Maggie, but Maggie prefers Zara over Leena.
  - ✓ Emma ↔ Maggie and Leena ↔ Zara: This is not stable since Emma prefers Leena over Maggie and Leena prefers Emma over Zara.
  - ✓ Emma ↔ Zara and Leena ↔ Maggie: This is not stable since Emma prefers Maggie over Zara and Maggie prefers Emma over Leena.

Stable Roomates

- **Example 5.18** Before the four women from Example 5.17 are paired as roommates, Maggie and Zara get into an argument, causing them to adjust their preference lists. Determine if a stable matching exists.

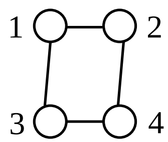
Emma:	l	>	m	>	z
Leena:	m	>	e	>	z
Maggie:	e	>	l	>	z
Zara:	e	>	l	>	m

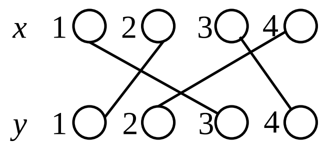
  - ✓ Emma ↔ Leena and Maggie ↔ Zara: This is not stable since Leena prefers Maggie over Emma and Maggie prefers Leena over Zara.
  - ✓ Emma ↔ Maggie and Leena ↔ Zara: This is not stable since Emma prefers Leena over Maggie and Leena prefers Emma over Zara.
  - ✓ Emma ↔ Zara and Leena ↔ Maggie: This is not stable since Emma prefers Maggie over Zara and Maggie prefers Emma over Leena.

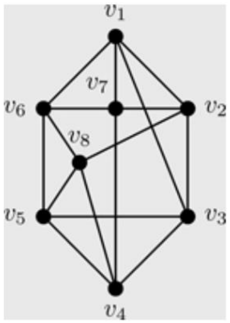
- In fact, the number of possible ways to pair n people (where n is even) is $(n-1)!!$, called *ndouble factorial*. For a given integer k, $k!!$ is defined as the product of all even integers less than or equal to k if k is even and the product of all odd integers less than or equal to k if k is odd.

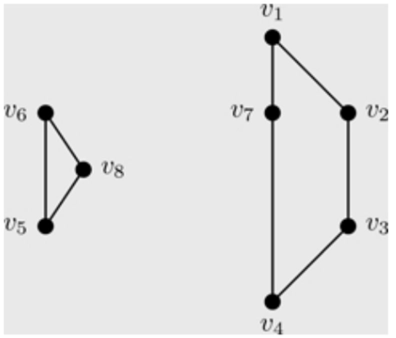
Factors

- **Definition 5.19** Let G be a graph with spanning subgraph H and let k be a positive integer. Then H is a k-factor of G if H is a k-regular.
- **Example 5.19** Find a 2-factor for the graph shown below.





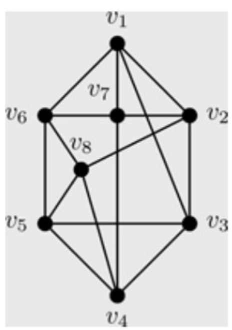


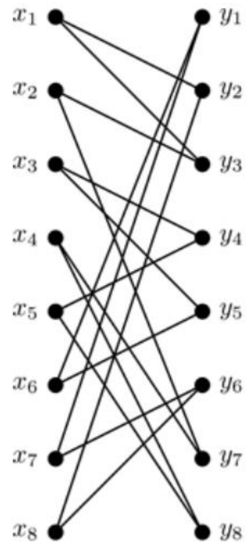


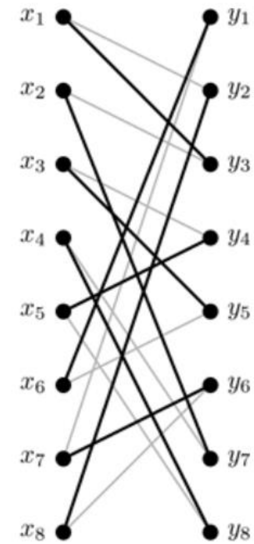
- **Theorem 5.20** If G is a $2k$-regular graph, then G has a 2-factor.
  - ✓ First, assume G is connected as otherwise apply the argument to each component of G.
  - ✓ Next, $\forall v \in G$, $deg(v)$ is even $\rightarrow$ contain an eulerian circuit C by Theorem 2.9.
  - ✓ Create a bipartite graph G' based upon this eulerian circuit and use a matching on G' to produce our 2-factor of G.
  - ✓ Let $V(G)=\{v_1, v_2, \ldots, v_n\}$ and define the vertices of G' as $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ so that x_iy_j is an edge of G' when v_i immediately precedes v_j on the eulerian circuit C.
  - ✓ G is $2k$-regular, C enters and exits each vertex of G exactly k times, and so G' is a k-regular bipartite graph $\rightarrow$ G' contains a perfect matching M by Corollary 5.8.
  - ✓ the edge incident to x_i in M represents an edge exiting v_i, whereas an edge incident to y_i represents an edge entering v_i.
  - ✓ Thus to find H, we start at v_1 and take the edge to v_i that arises from the matched edge x_1y_i. The next edge in the 2-factor will be from v_i to v_j arising from the matched edge x_iy_j.
  - ✓ Thus will continue until all vertices are listed, creating a 2-regular spanning subgraph of G.

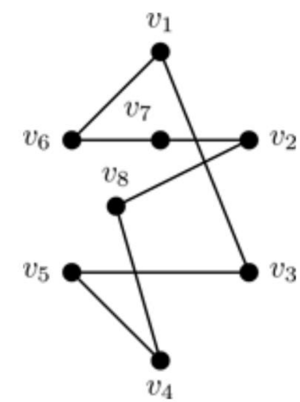
Factors

- **Example 5.19**
  - ✓ C: $v_1v_3v_5v_4v_8v_2v_7v_6v_5v_8v_6v_1v_2v_3v_4v_7v_1$

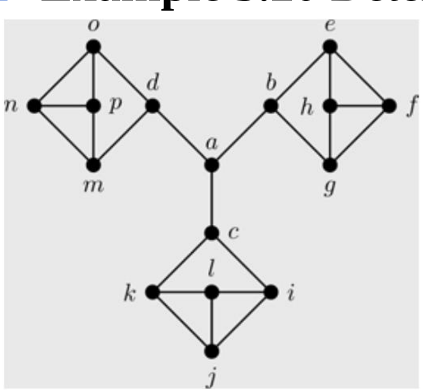


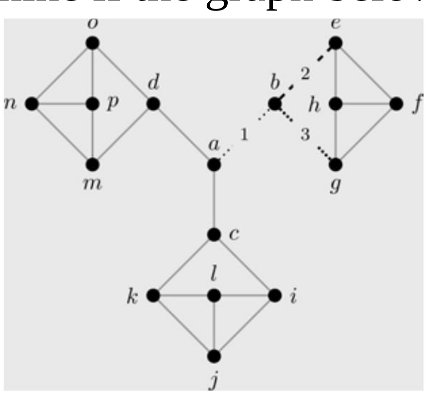


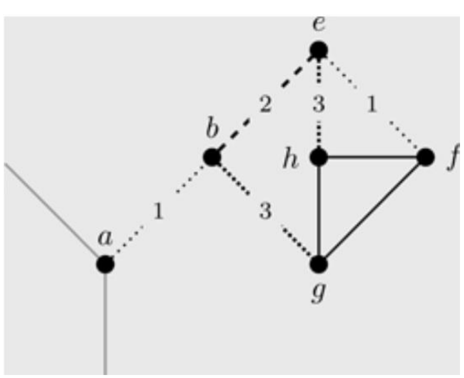


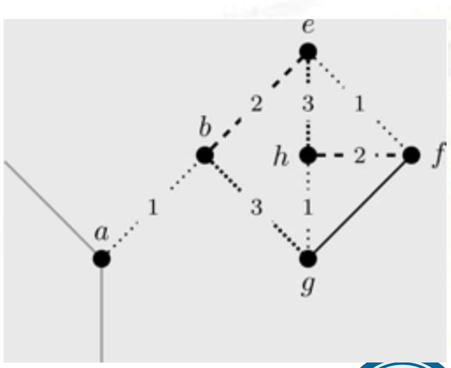


- **Definition 5.21** A *k-factorization* of G is a partition of the edges into disjoint k-factors.
- **Example 5.20** Determine if the graph below has a 1-factorization.









Factors

- **Proposition 5.22** Every k-regular bipartite graph has a 1-factorization for all $k \geq 1$.

- **Theorem 5.23** A graph G has a 2-factorization if and only if G is $2k$-regular.