Midterm Review

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS445/index.html

Exam code

- Exam on Oct 29 10:00-12:00 at Dong Shang Yuan 407 (lecture classroom)
- Finish the exam paper by yourself
- Allowed:
  - Calculator, watch (not smart)
- Not allowed:
  - Books, materials, cheat sheet, …
  - Phones, any smart device
- No entering after 10:30
- Early submission period: 10:30--

Grading policy

- Attendance and participance: 5%
- Assignments: 35%
- Midterm exam: 20%
- Project: 10%
- Final exam: 30%

Covered topics

- Basics
  - Graphs, paths/walks/cycles, bipartite graphs
- Connectivity
- Trees
- Circuits
- Matchings

Basic Concepts

Graphs

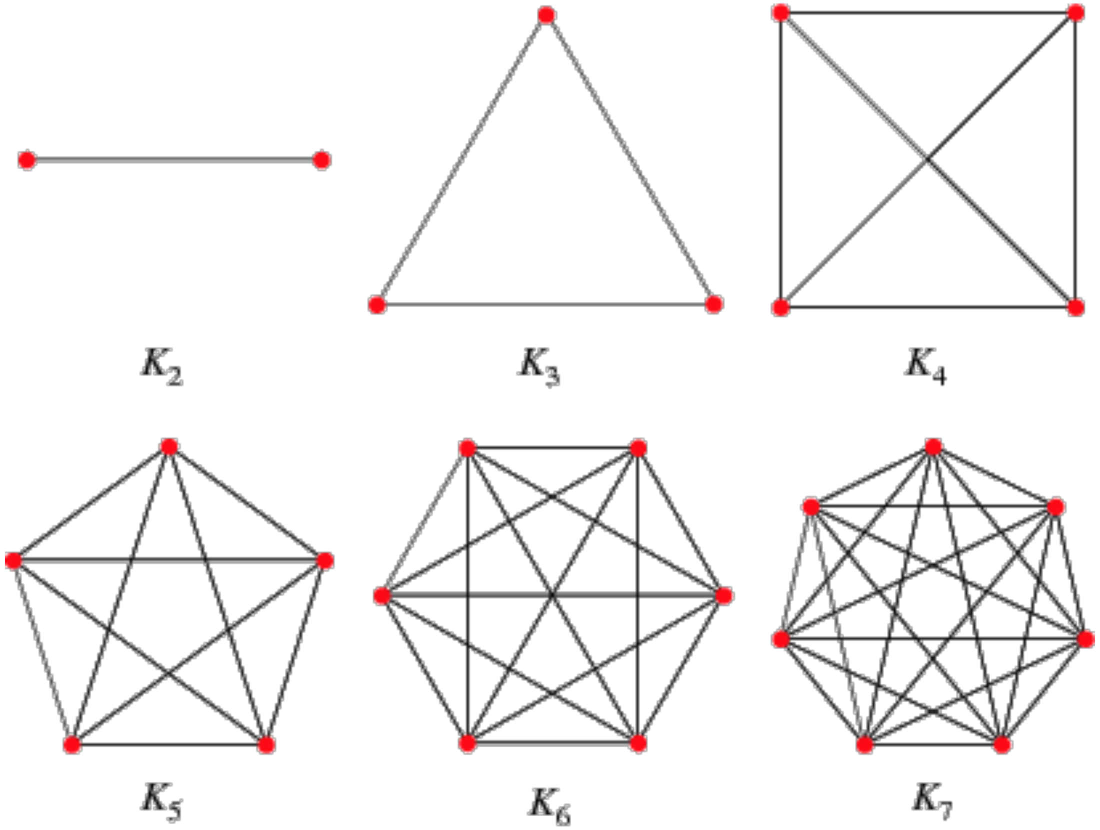
We mainly focus on
Simple graph:
No loops, no multi-edges

- Definition A graph G is a pair (V, E)
  - V: set of vertices
  - E: set of edges
  - $e \in E$ corresponds to a pair of endpoints $x, y \in V$
- Two graphs $G_1 = (V_1, E_1), G_1 = (V_2, E_2)$ are isomorphic if there is a bijection $f: V_1 \to V_2$ s.t.

$$e = \{a, b\} \in E_1 \Leftrightarrow f(e) := \{f(a), f(b)\} \in E_2$$

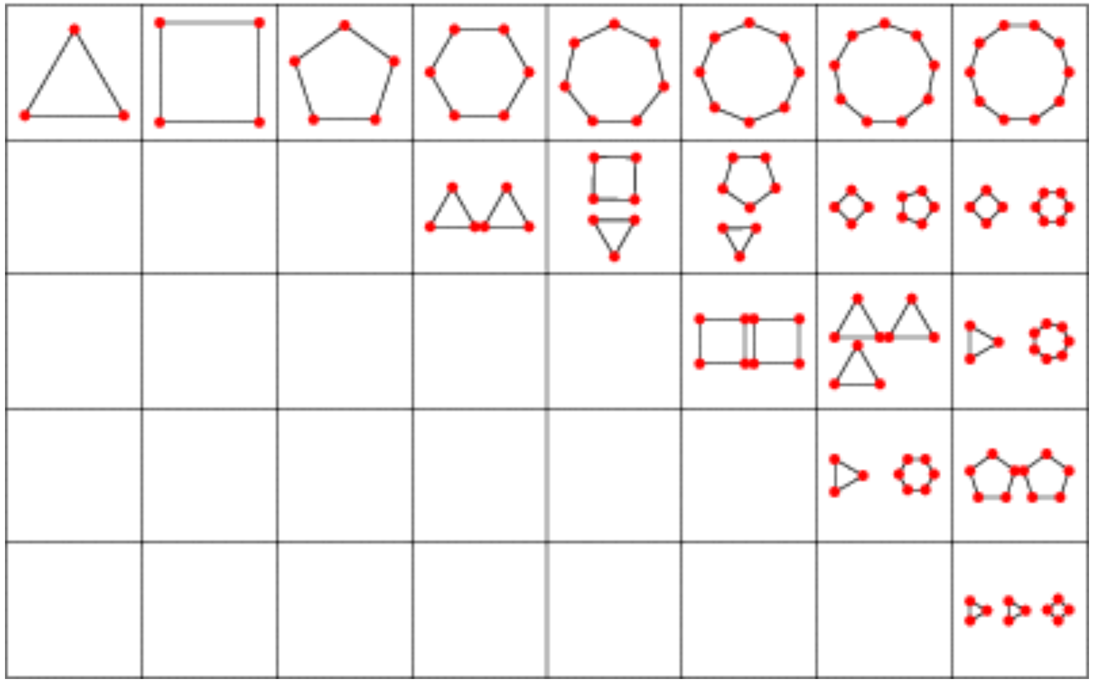
Example: Complete graphs

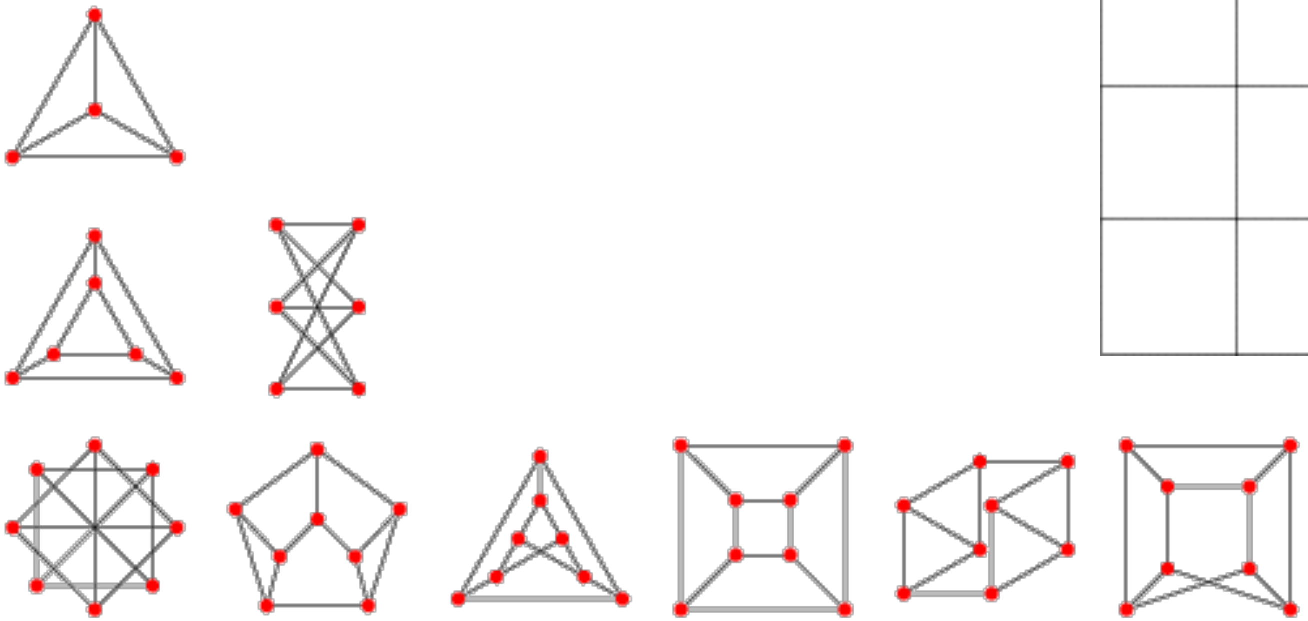
- There is an edge between every pair of vertices



Example: Regular graphs

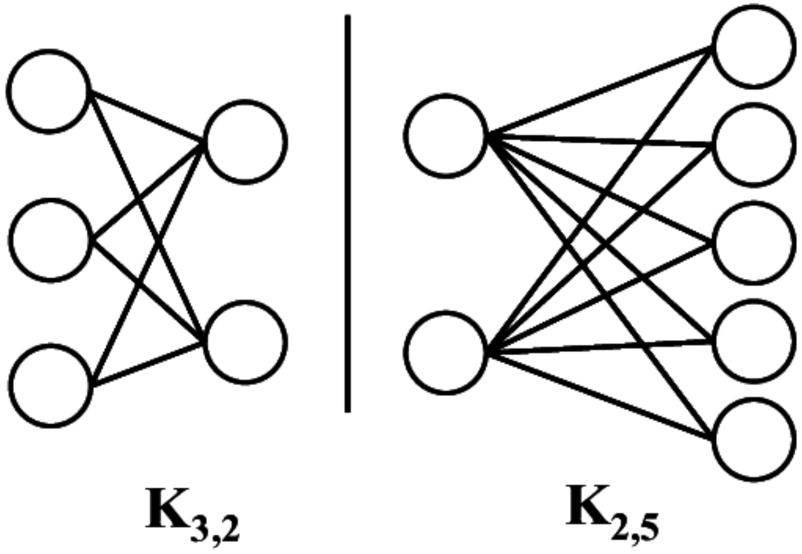
- Every vertex has the same degree





Example: Bipartite graphs

- The vertex set can be partitioned into two sets X and Y such that every edge in G has one end vertex in X and the other in Y
- Complete bipartite graphs



Example (1A, L): Peterson graph

- Show that the following two graphs are same/isomorphic

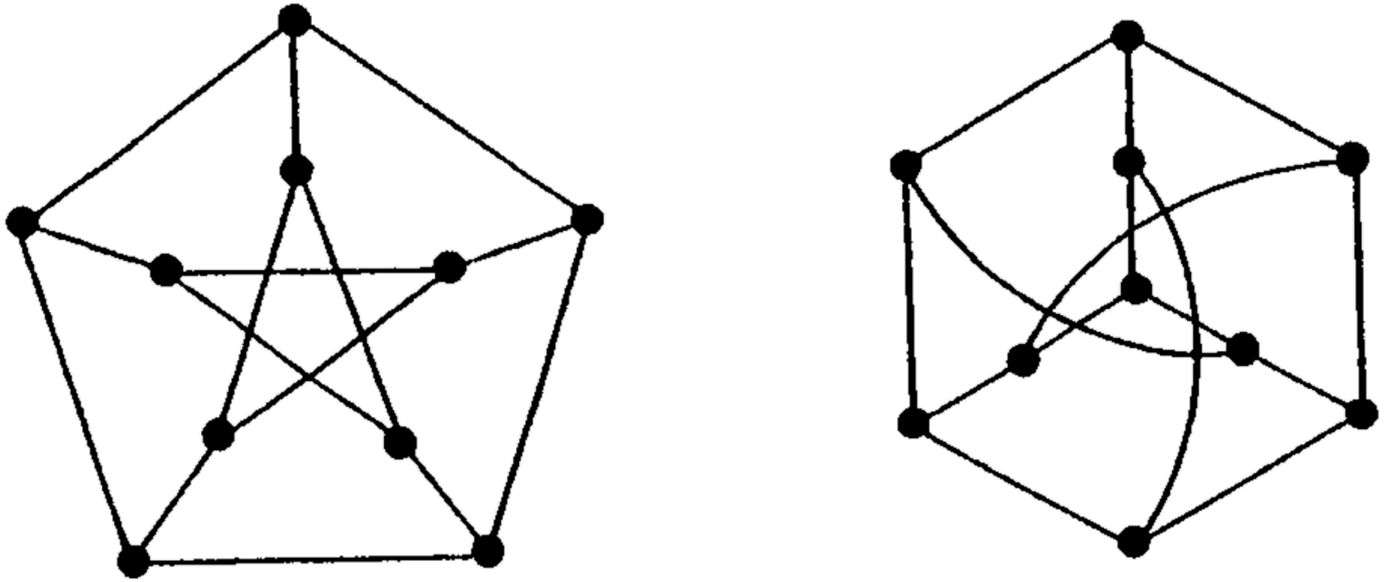


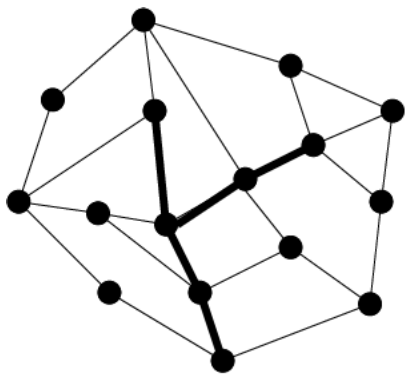
Figure 1.4

Subgraphs

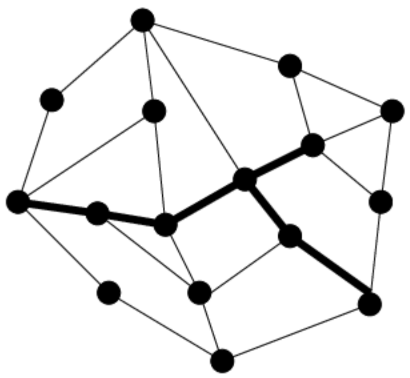
- A subgraph of a graph G is a graph H such that

$$V(H) \subseteq V(G), E(H) \subseteq E(G)$$

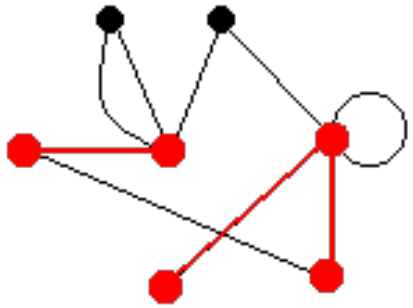
  and the ends of an edge $e \in E(H)$ are the same as its ends in G
  - H is a spanning subgraph when $V(H) = V(G)$
  - The subgraph of G induced by a subset $S \subseteq V(G)$ is the subgraph whose vertex set is S and whose edges are all the edges of G with both ends in S



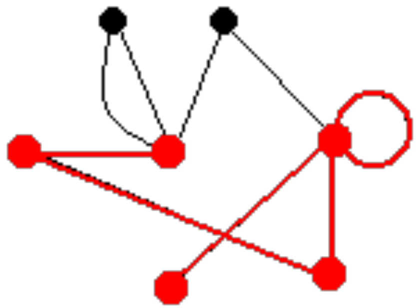
(a)



(b)



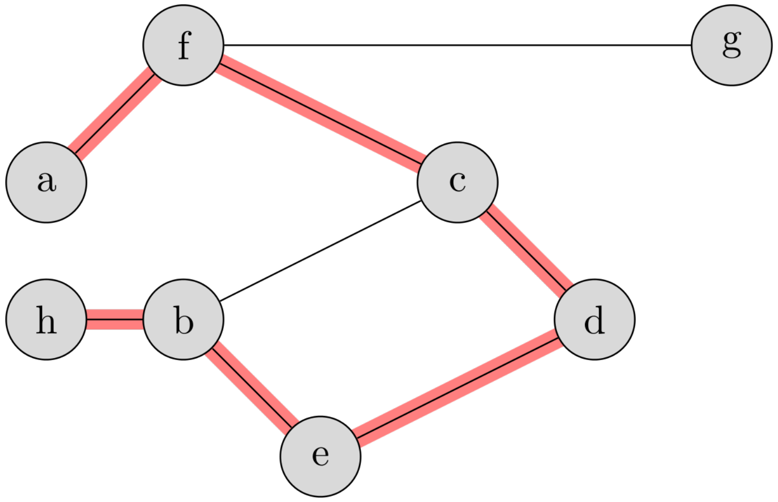
Subgraph (in red)



Induced Subgraph

Paths (路径)

- A path is a nonempty graph $P = (V, E)$ of the form
$$V = \{x_0, x_1, \dots, x_k\} \quad E = \{x_0x_1, x_1x_2, \dots, x_{k-1}x_k\}$$
where the x_i are all distinct
- P^k: path of length k (the number of edges)

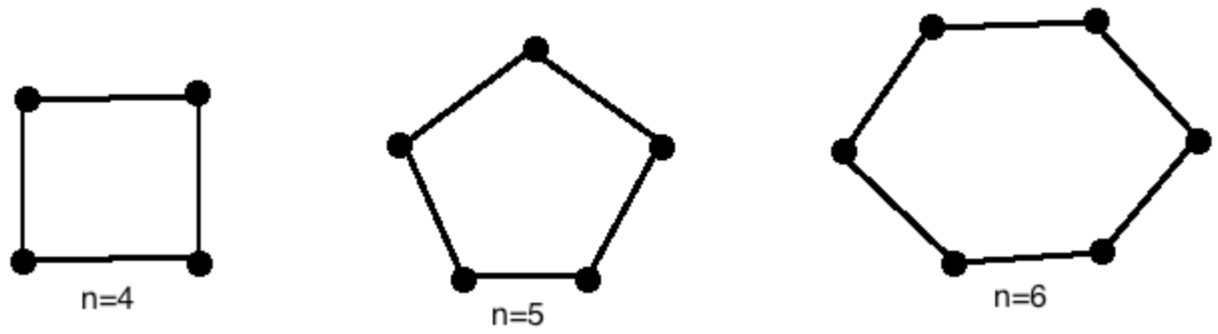


Walk (游走)

- A walk is a non-empty alternating sequence $v_0 e_1 v_1 e_2 \dots e_k v_k$
  - The vertices not necessarily distinct
  - The length = the number of edges
- Proposition (1.2.5, W) Every u-v walk contains a u-v path

Cycles (环)

- If $P = x_0x_1 \ldots x_{k-1}$ is a path and $k \geq 3$, then the graph $C \coloneqq P + x_{k-1}x_0$ is called a cycle
- C^k: cycle of length k (the number of edges/vertices)



- Proposition (1.2.15, W) Every closed odd walk contains an odd cycle

Neighbors and degree

- Two vertices $a \neq b$ are called adjacent if they are joined by an edge
  - $N(x)$: set of all vertices adjacent to x
    - neighbors of x
  - A vertex is isolated vertex if it has no neighbors

Handshaking Theorem (Euler 1736)

- Theorem A finite graph G has an even number of vertices with odd degree.
- Proof The degree of x is the number of times it appears in the right column. Thus

$$\sum_{x \in V(G)} \deg(x) = 2|E(G)|$$

edge	ends
a	x, z
b	y, w
c	x, z
d	z, w
e	z, w
f	x, y
g	z, w

Figure 1.1

Degree

- Minimal degree of G: $\delta(G) = \min\{d(v): v \in V\}$
- Maximal degree of G: $\Delta(G) = \min\{d(v): v \in V\}$
- Average degree of G: $d(G) = \frac{1}{|V|} \sum_{v \in V} d(v) = \frac{2|E|}{|V|}$

- All measures the `density' of a graph

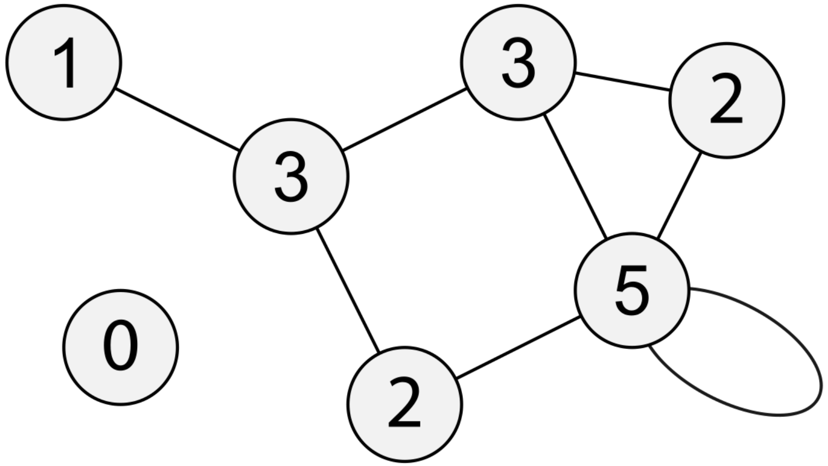
- $d(G) \geq \delta(G)$

Degree (global to local)

- Proposition (1.2.2, D) Every graph G with at least one edge has a subgraph H with

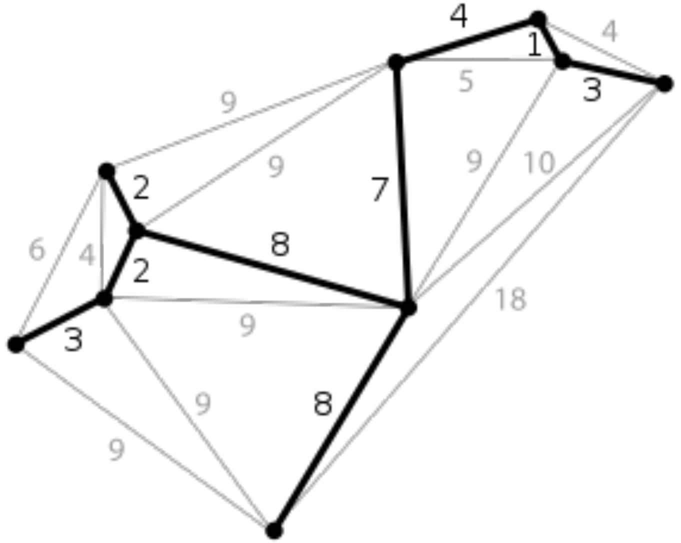
$$\delta(H) > \frac{1}{2}d(H) \geq \frac{1}{2}d(G)$$

- Example: $|G| = 7, d(G) = 16/7$



Minimal degree guarantees long paths and cycles

- Proposition (1.3.1, D) Every graph G contains a path of length $\delta(G)$ and a cycle of length at least $\delta(G)+1$, provided $\delta(G) \geq 2$.



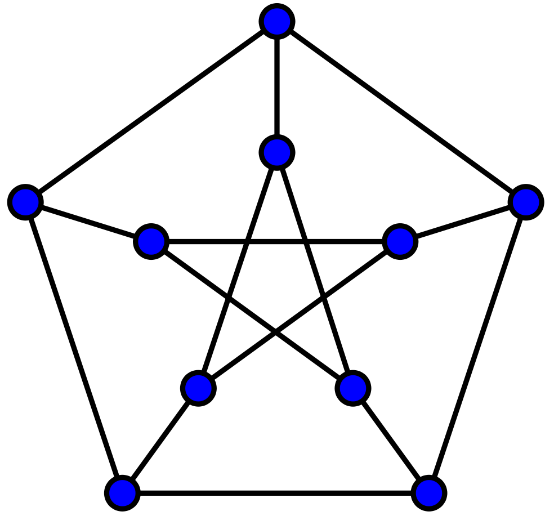
Distance and diameter

- The distance $d_G(x, y)$ in G of two vertices x, y is the length of a shortest $x \sim y$ path
  - if no such path exists, we set $d(x, y) \coloneqq \infty$
- The greatest distance between any two vertices in G is the diameter of G

Girth

- The minimum length of a cycle in a graph G is the girth $g(G)$ of G

- Example: The Peterson graph is the unique 5-cage
  - cubic graph (every vertex has degree 3)
  - girth = 5
  - smallest graph satisfies the above properties
- A tree has girth ∞



Girth and diameter/minimal degree

- Proposition (1.3.2, D) Every graph G containing a cycle satisfies $g(G) \leq 2\,\mathrm{diam}(G) + 1$
- $n_0(\delta, g) \coloneqq \begin{cases} 1 + \delta \sum_{i=0}^{r-1} (\delta - 1)^i\,, & \text{if } g = 2r+1 \text{ is odd} \\ 2 \sum_{i=0}^{r-1} (\delta - 1)^i\,, & \text{if } g = 2r \text{ is even} \end{cases}$
- Exercise (Ex7, ch1, D) Let G be a graph. If $\delta(G) \geq \delta \geq 2$ and $g(G) \geq g$, then $|G| \geq n_0(\delta, g)$
- Corollary (1.3.5, D) If $\delta(G) \geq 3$, then $g(G) < 2\log|G|$

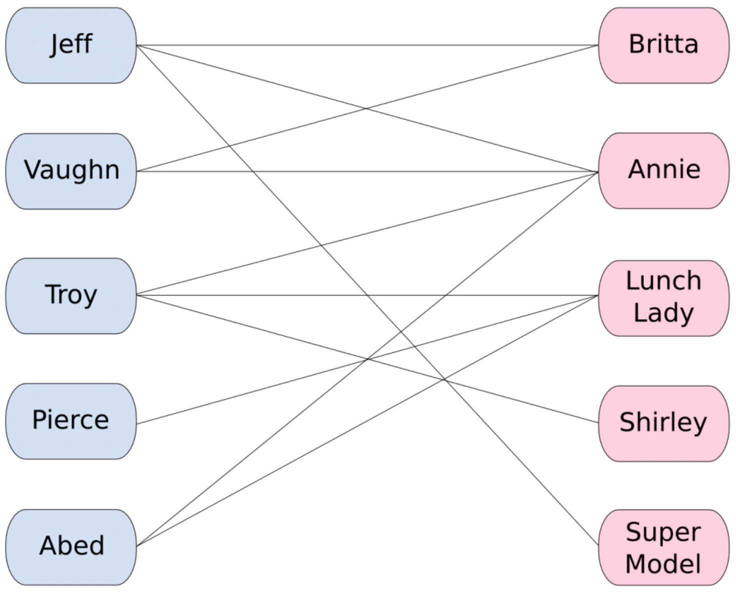
Triangle-free bounds # of edges

- Theorem (1.3.23, W, Mantel 1907) The maximum number of edges in an n-vertex triangle-free simple graph is $\lfloor n^2/4 \rfloor$

- The bound is best possible
- There is a triangle-free graph with $\lfloor n^2/4 \rfloor$ edges: $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$
- Extremal problems

Bipartite graphs

- Theorem (1.2.18, W, Kőnig 1936)
  A graph is bipartite $\Leftrightarrow$ it contains no odd cycle



Proposition (1.2.15, W) Every closed odd walk contains an odd cycle

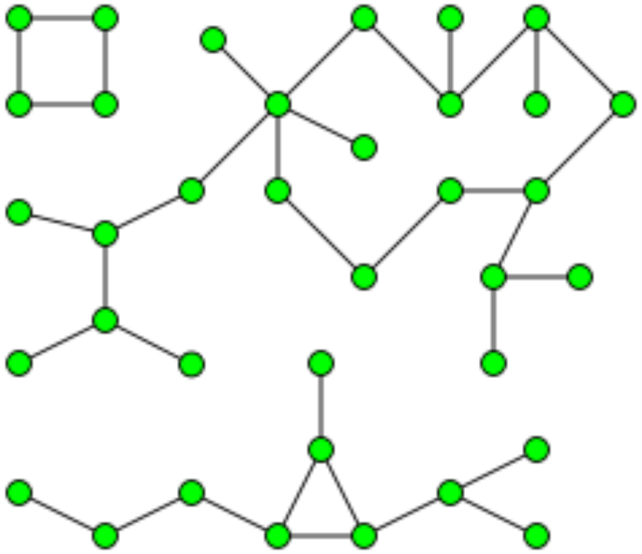
Complete graph is a union of bipartite graphs

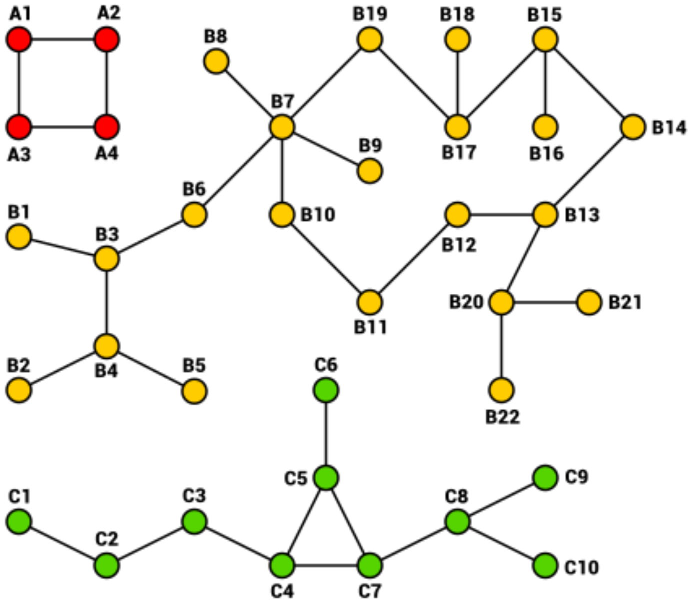
- The union of graphs $G_1, \ldots, G_k$, written $G_1 \cup \cdots \cup G_k$, is the graph with vertex set $\bigcup_{i=1}^{k} V(G_i)$ and edge set $\bigcup_{i=1}^{k} E(G_i)$
- Theorem (1.2.23, W) The complete graph K_n can be expressed as the union of k bipartite graphs $\Leftrightarrow n \leq 2^k$
- Theorem (1.3.19, W) Every loopless graph G has a bipartite subgraph with at least $|E|/2$ edges

Connectivity

Connected, connected component

- A graph G is connected if $G \neq \emptyset$ and any two of its vertices are linked by a path
- A maximal connected subgraph of G is a (connected) component





Quiz

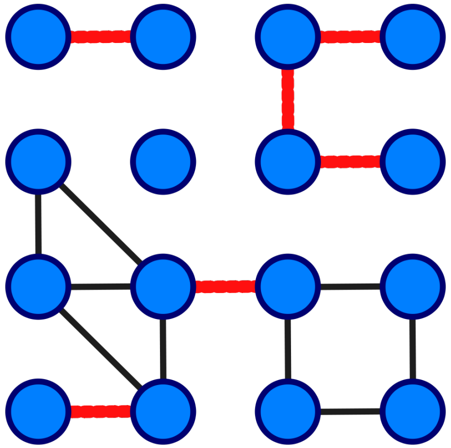
- Problem (1B, L) Suppose G is a graph on 10 vertices that is not connected. Prove that G has at most 36 edges. Can equality occur?
- More general (Ex9, S1.1.2, H) Let G be a graph of order n that is not connected. What is the maximum size of G?

Connected vs. minimal degree

- Proposition (1.3.15, W) If $\delta(G) \geq \frac{n-1}{2}$, then G is connected
- (Ex16, S1.1.2, H) (1.3.16, W)
  If $\delta(G) \geq \frac{n-2}{2}$, then G need not be connected

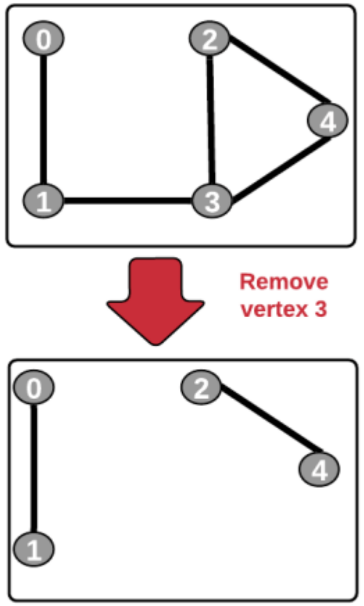
- Extremal problems
- "best possible" "sharp"

Add/delete an edge



- Components are pairwise disjoint; no two share a vertex
- Adding an edge decreases the number of components by 0 or 1
  - $\Rightarrow$ deleting an edge increases the number of components by 0 or 1
- Proposition (1.2.11, W)
  Every graph with n vertices and k edges has at least $n - k$ components
- An edge e is called a bridge if the graph $G - e$ has more components
- Proposition (1.2.14, W)
  An edge e is a bridge $\Leftrightarrow$ e lies on no cycle of G
  - Or equivalently, an edge e is not a bridge $\Leftrightarrow$ e lies on a cycle of G

Cut vertex and connectivity



- A node v is a cut vertex if the graph $G - v$ has more components
- A proper subset S of vertices is a vertex cut set if the graph $G - S$ is disconnected
- The connectivity, $\kappa(G)$, is the minimum size of a cut set of G
  - The graph is k-connected for any $k \leq \kappa(G)$

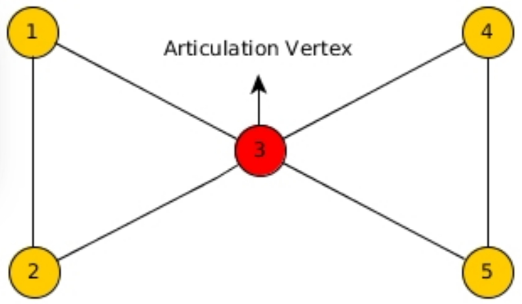
Connectivity properties

- $\kappa(K^n) := n - 1$
- If G is disconnected, $\kappa(G) = 0$
  - $\Rightarrow$ A graph is connected $\Leftrightarrow \kappa(G) \geq 1$
- If G is connected, non-complete graph of order n, then

$$1 \leq \kappa(G) \leq n - 2$$

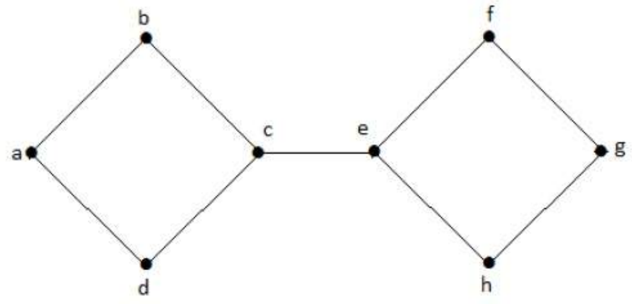
Connectivity properties (cont.)

Proposition (1.2.14, W)
An edge e is a bridge $\Leftrightarrow$ e lies on no cycle of G
- Or equivalently, an edge e is not a bridge $\Leftrightarrow$ e lies on a cycle of G



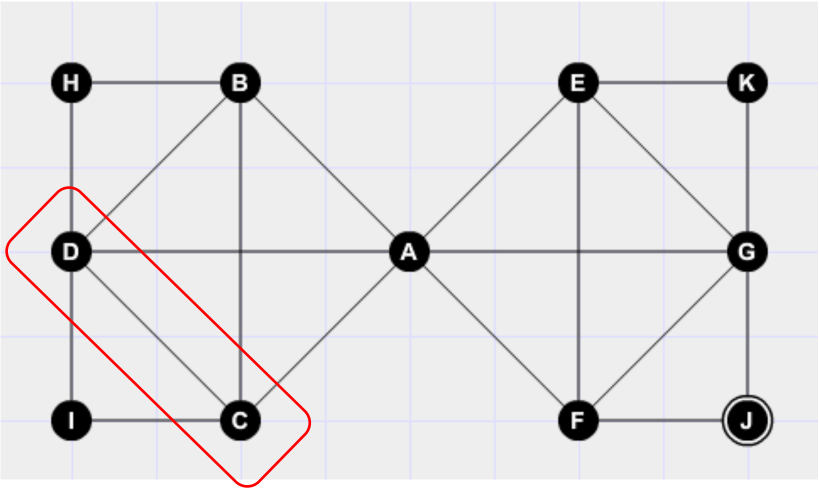
- $\kappa(G) \geq 2 \Leftrightarrow G$ is connected and has no cut vertices
- A vertex lies on a cycle $\not\Rightarrow$ it is not a cut vertex
  - $\Rightarrow$ (Ex13, S1.1.2, H) Every vertex of a connected graph G lies on at least one cycle $\not\Rightarrow \kappa(G) \geq 2$
  - (Ex14, S1.1.2, H) $\kappa(G) \geq 2$ implies G has at least one cycle

- (Ex12, S1.1.2, H) G has a cut vertex vs. G has a bridge



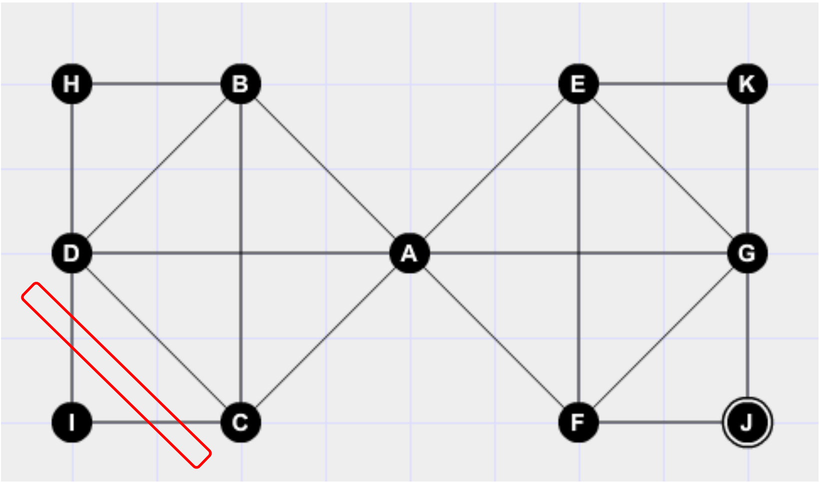
Connectivity and minimal degree

- (Ex15, S1.1.2, H)
- $\kappa(G) \leq \delta(G)$
- If $\delta(G) \geq n - 2$, then $\kappa(G) = \delta(G)$



Edge-connectivity

- A proper subset $F \subset E$ is edge cut set if the graph $G - F$ is disconnected
- The edge-connectivity $\lambda(G)$ is the minimal size of edge cut set
- $\lambda(G) = 0$ if G is disconnected
- Proposition (1.4.2, D) If G is non-trivial, then $\kappa(G) \leq \lambda(G) \leq \delta(G)$



Trees

Definition and properties

- A tree is a connected graph T with no cycles
- Recall that a graph is bipartite $\Leftrightarrow$ it has no odd cycle
- (Ex 3, S1.3.1, H) A tree of order $n \geq 2$ is a bipartite graph

- Recall that an edge e is a bridge $\Leftrightarrow$ e lies on no cycle of G
- $\Rightarrow$ Every edge in a tree is a bridge
- T is a tree $\Leftrightarrow$ T is minimally connected, i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$

Equivalent definitions (Theorem 1.5.1, D)

- T is a tree of order n
  $\Leftrightarrow$ Any two vertices of T are linked by a unique path in T
  $\Leftrightarrow$ T is minimally connected
  - i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$

  $\Leftrightarrow$ T is maximally acyclic
  - i.e. T contains no cycle but $T + xy$ does for any non-adjacent vertices $x, y \in T$

  $\Leftrightarrow$ (Theorem 1.10, 1.12, H) T is connected with $n - 1$ edges

  $\Leftrightarrow$ (Theorem 1.13, H) T is acyclic with $n - 1$ edges

Leaves of tree

- A vertex of degree 1 in a tree is called a leaf
- Theorem (1.14, H; Ex9, S1.3.2, H) Let T be a tree of order $n \geq 2$. Then T has at least two leaves
- (Ex3, S1.3.2, H) Let T be a tree with max degree Δ. Then T has at least Δ leaves
- (Ex10, S1.3.2, H) Let T be a tree of order $n \geq 2$. Then the number of leaves is
$$2 + \sum_{v:d(v)\geq 3} (d(v) - 2)$$
- (Ex8, S1.3.2, H) Every nonleaf in a tree is a cut vertex

Properties

- The center of a tree
- Theorem (1.15, H) In any tree, the center is either a single vertex or a pair of adjacent vertices
- Tree as subgraphs
- Theorem (1.16, H) Let T be a tree of order $k+1$ with k edges. Let G be a graph with $\delta(G) \geq k$. Then G contains T as a subgraph

Spanning tree

- Given a graph G and a subgraph T, T is a spanning tree of G if T is a tree that contains every vertex of G
- Example: A telecommunications company tries to lay cable in a new neighbourhood
- Proposition (2.1.5c, W) Every connected graph contains a spanning tree

Minimal spanning tree - Kruskal's Algorithm

- Given: A connected, weighted graph G

1. Find an edge of minimum weight and mark it.
2. Among all of the unmarked edges that do not form a cycle with any of the marked edges, choose an edge of minimum weight and mark it
3. If the set of marked edges forms a spanning tree of G, then stop. If not, repeat step 2

Example

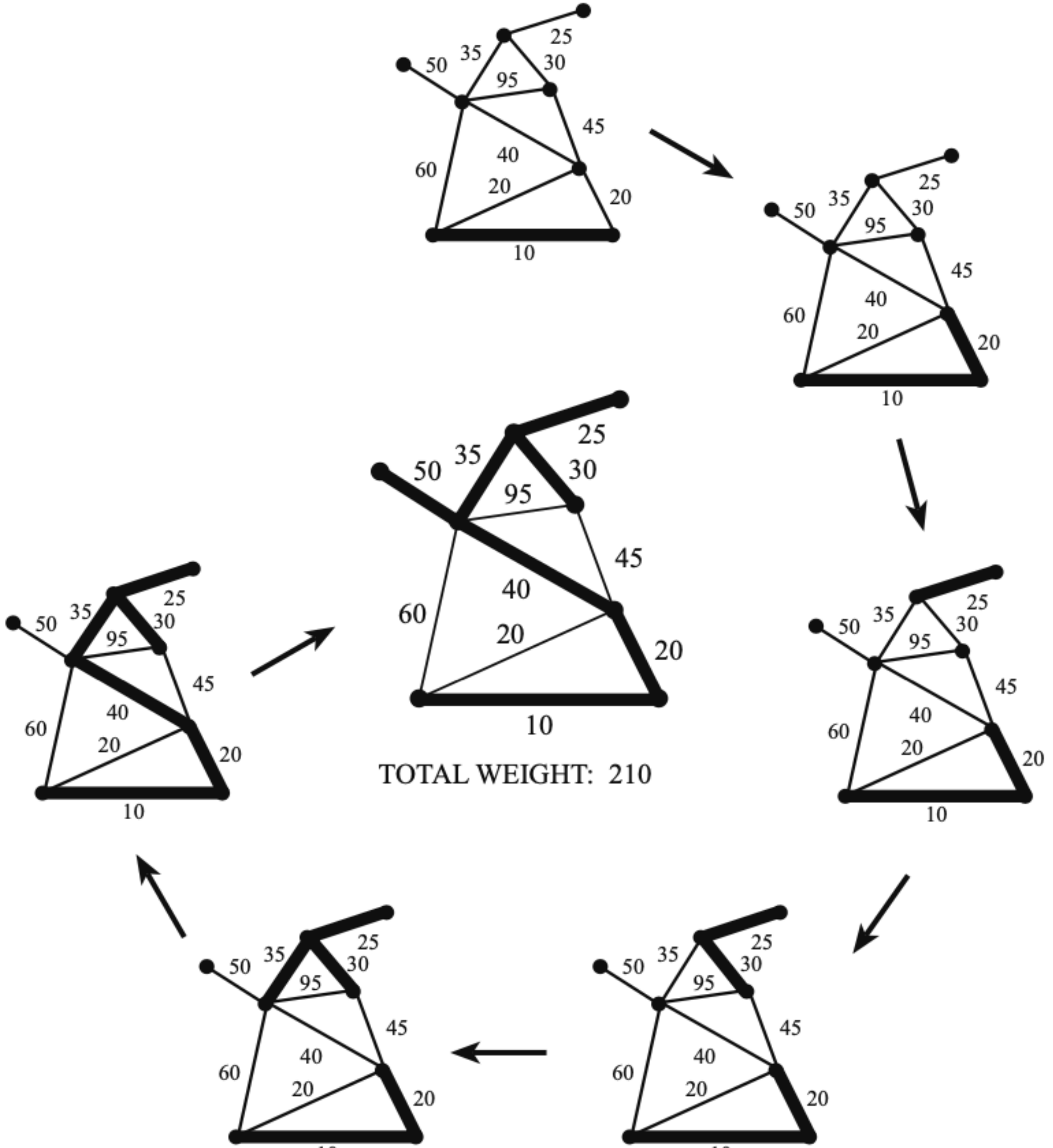


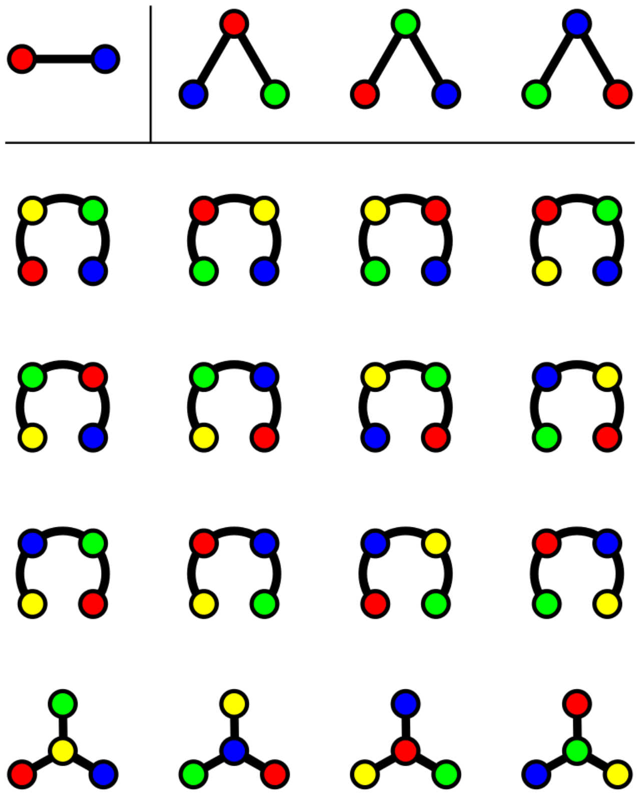
FIGURE 1.43. The stages of Kruskal's algorithm.

Theoretical guarantee of Kruskal’s algorithm

- Theorem (1.17, H) Kruskal’s algorithm produces a spanning tree of minimum total weight

Cayley’s tree formula

- Theorem (1.18, H). There are n^{n-2} distinct labeled trees of order n



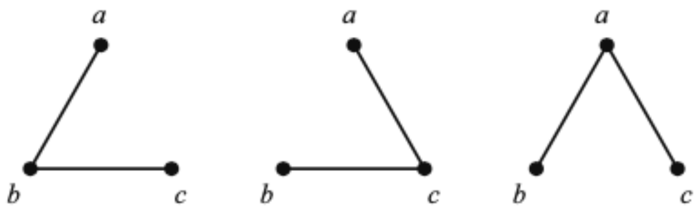


FIGURE 1.45. Labeled trees on three vertices.

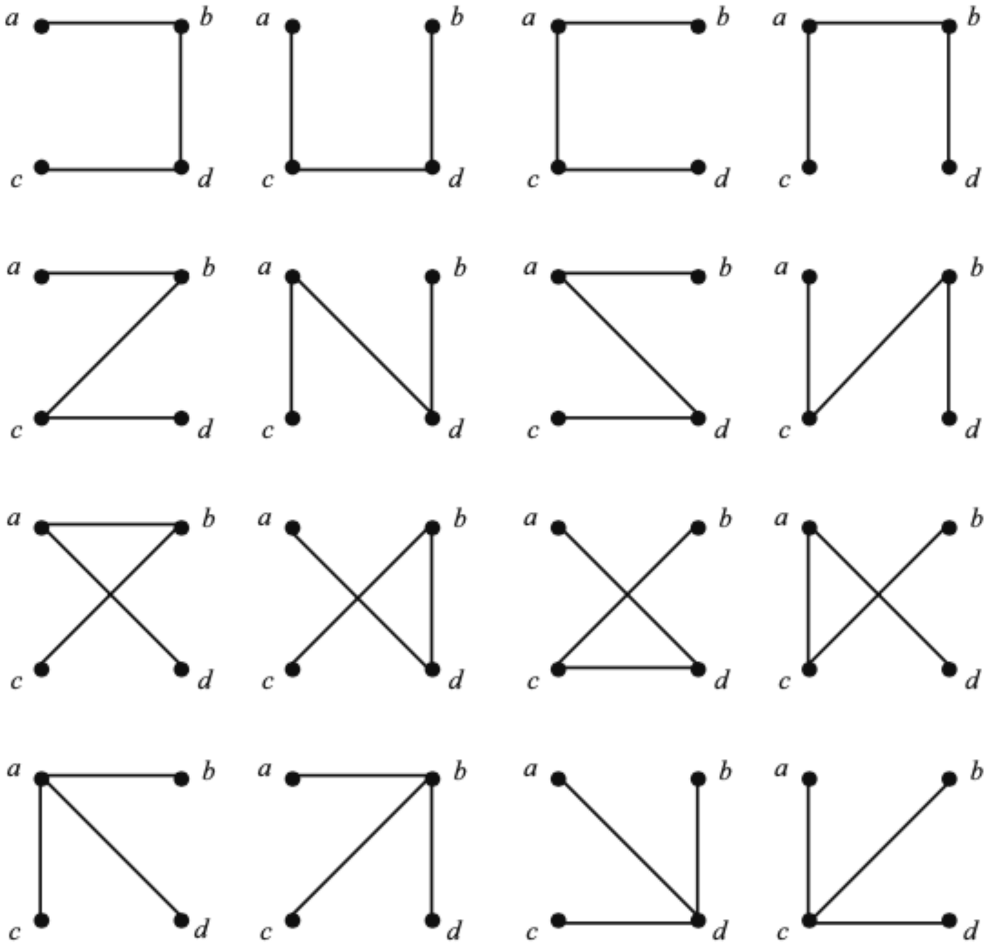


FIGURE 1.46. Labeled trees on four vertices.

Example

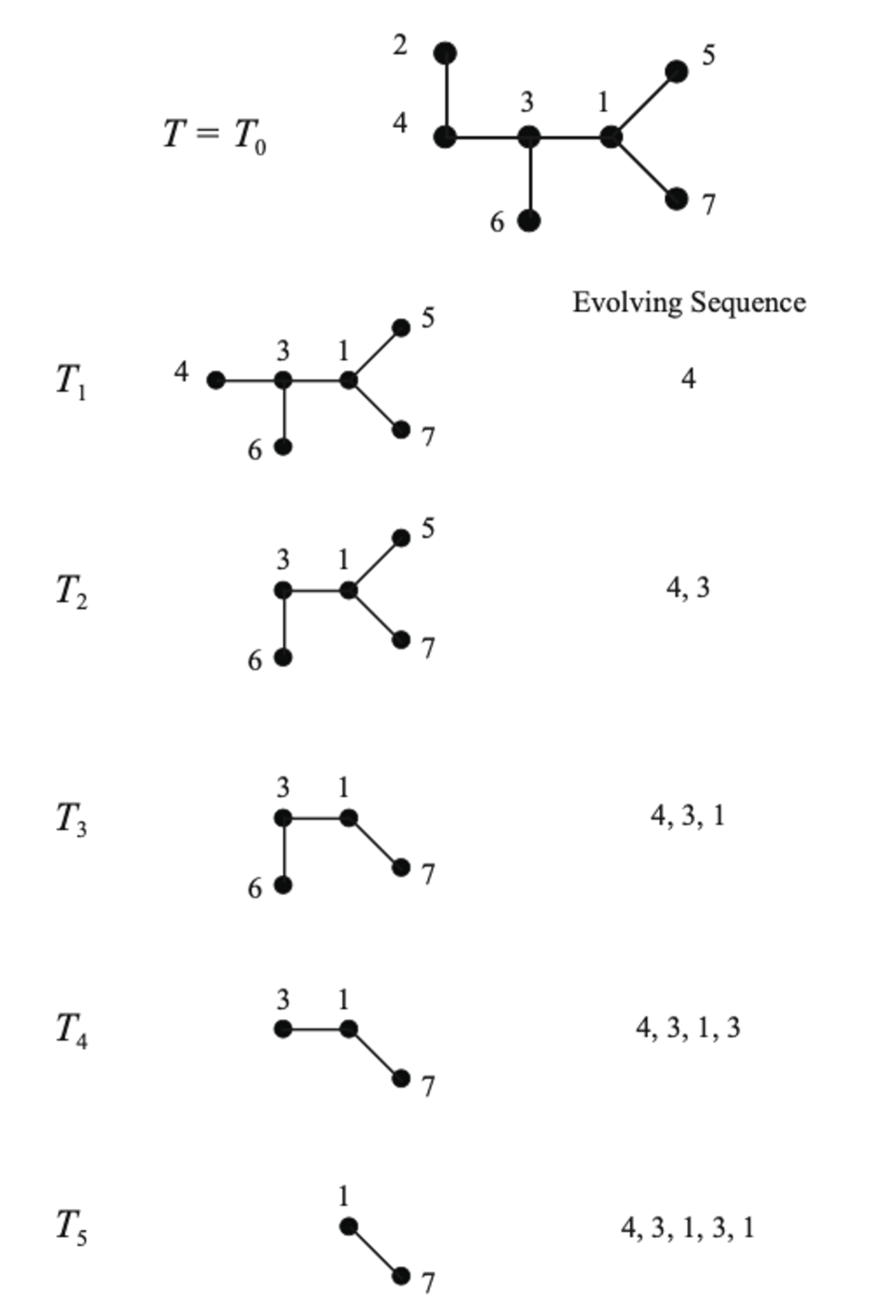


FIGURE 1.47. Creating a Prüfer sequence.

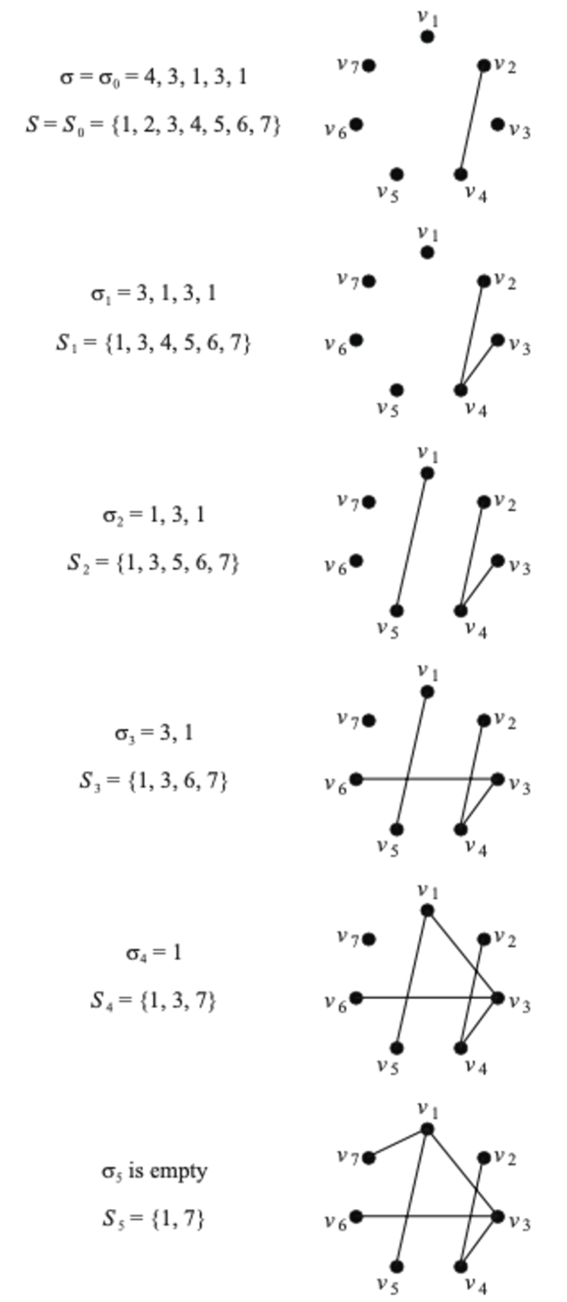
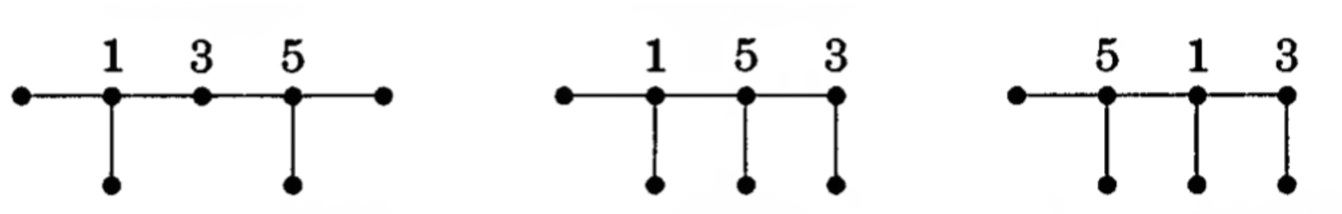


FIGURE 1.48. Building a labeled tree.

Trees with fixed degrees

- Corollary (2.2.4, W) Given positive integers $d_1, \ldots, d_n$ summing to $2n-2$, there are exactly $\frac{(n-2)!}{\prod(d_i-1)!}$ trees with vertex set $[n]$ such that vertex i has degree d_i for each i

- Example (2.2.5, W) Consider trees with vertices $[7]$ that have degrees $(3,1,2,1,3,1,1)$



Matrix tree theorem - cofactor

- For an $n \times n$ matrix A, the i, j cofactor of A is defined to be
$$(-1)^{i+j} \det(M_{ij})$$
where M_{ij} represents the $(n-1) \times (n-1)$ matrix formed by deleting row i and column j from A

3 × 3 generic matrix [edit]

Consider a 3×3 matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Its cofactor matrix is

$$\mathbf{C} = \begin{pmatrix} +\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} \\ +\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} & +\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{pmatrix},$$

Matrix tree theorem

- Theorem (1.19, H; 2.2.12, W; Kirchhoff) If G is a connected labeled graph with adjacency matrix A and degree matrix D, then the number of unique spanning trees of G is equal to the value of any cofactor of the matrix $D - A$
- If the row sums and column sums of a matrix are all 0, then the cofactors all have the same value
- Exercise (Ex7, S1.3.4, H) Use the matrix tree theorem to prove Cayley's theorem

Example

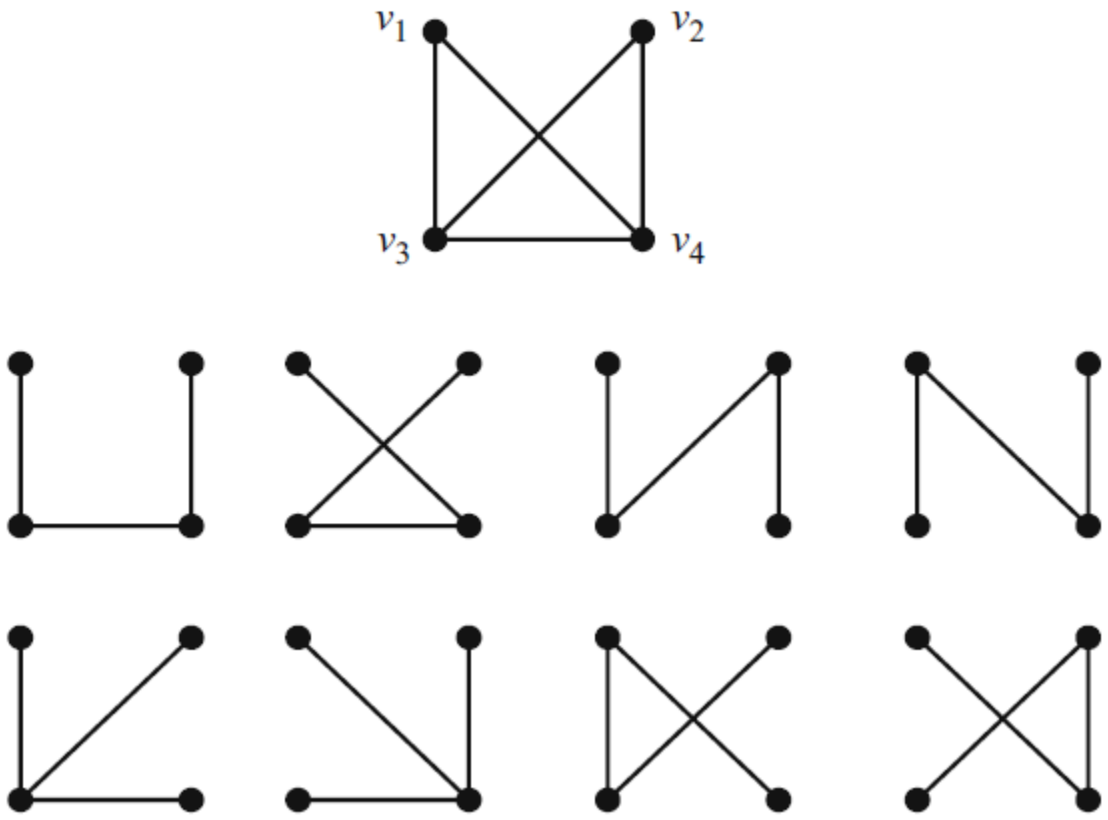


FIGURE 1.49. A labeled graph and its spanning trees.

The degree matrix D and adjacency matrix A are

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

and so

$$D - A = \begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}.$$

The $(1, 1)$ cofactor of $D - A$ is

$$\det \begin{bmatrix} 2 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix} = 8.$$

Score one for Kirchhoff!

Wiener index

- In a communication network, large diameter may be acceptable if most pairs can communicate via short paths. This leads us to study the average distance instead of the maximum
- Wiener index $D(G) = \sum_{u,v \in V(G)} d_G(u,v)$
- Theorem (2.1.14, W) Among trees with n vertices, the Wiener index $D(T)$ is minimized by stars and maximized by paths, both uniquely
- Over all connected n-vertex graphs, $D(G)$ is minimized by K_n and maximized by paths
  - (Corollary 2.1.16, W) If G is a connected n-vertex graph, then $D(G) \leq D(P_{n-1})$
    - (Lemma 2.1.15, W) If H is a subgraph of G, then $d_G(u,v) \leq d_H(u,v)$

Circuits

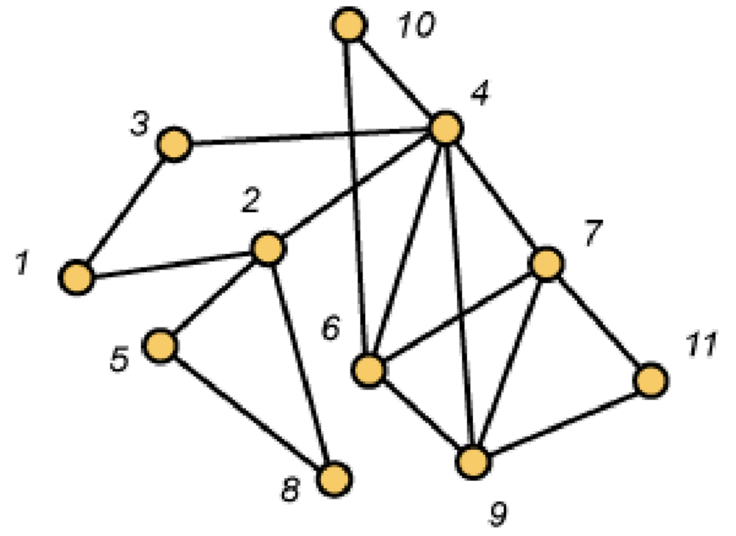
Eulerian circuit

- A closed walk through a graph using every edge once is called an Eulerian circuit
- A graph that has such a walk is called an Eulerian graph
- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree
- (possibly with multiple edges)
- Proof "$\Longrightarrow$" That G must be connected is obvious.
  Since the path enters a vertex through some edge and leaves by another edge, it is clear that all degrees must be even
  - Lemma (1.2.25, W) If every vertex of a graph G has degree at least 2, then G contains a cycle.

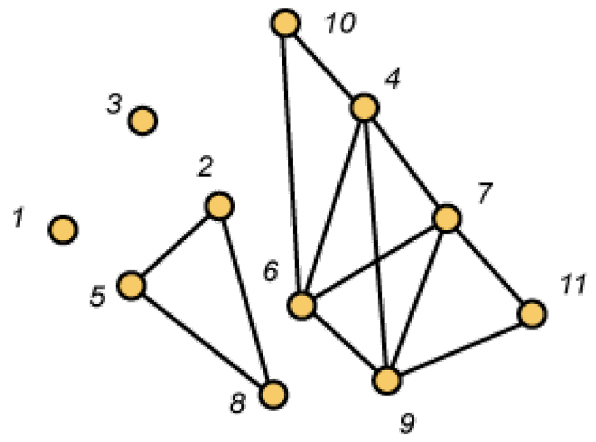
Hierholzer's Algorithm for Euler Circuits

1. Choose a root vertex r and start with the trivial partial circuit (r)
2. Given a partial circuit $(x_0, e_1, x_1, \dots, x_{t-1}, e_t, x_t = x_0)$ that traverses not all edges of G, remove these edges from G
3. Let i be the least integer for which x_i is incident with one of the remaining edges
4. Form a greedy partial circuit among the remaining edges of the form $(x_i = y_0, e'_1, y_1, \dots, y_{s-1}, e'_s, y_s = x_i)$
5. Expand the original circuit by setting $(x_0, e_1, \dots, e_i, x_i = y_0, e'_1, y_1, \dots, y_{s-1}, e'_s, y_s = x_i, e_{i+1}, \dots, e_t, x_t = x_0)$
6. Repeat step 2-5

Example

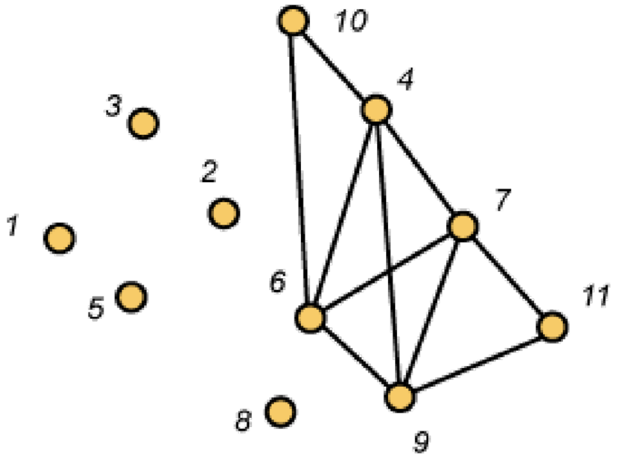


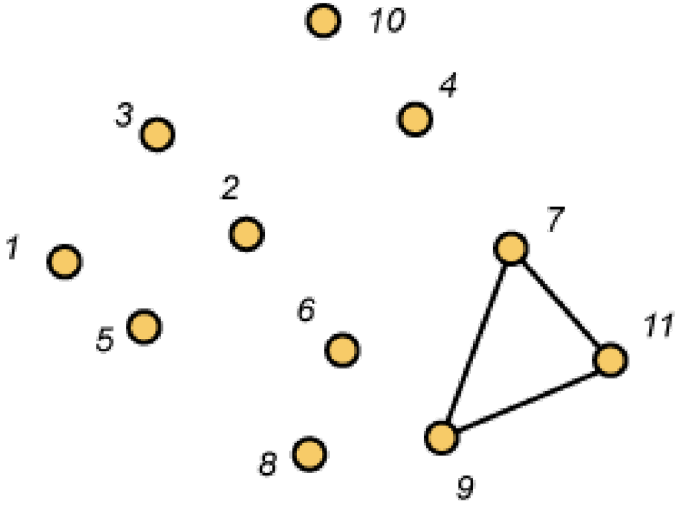
1. Start with the trivial circuit (1)
2. Greedy algorithm yields the partial circuit
   $(1,2,4,3,1)$
3. Remove these edges
4. The first vertex incident with remaining edges is 2
5. Greedy algorithms yields $(2,5,8,2)$
6. Expanding $(1,2,5,8,2,4,3,1)$
7. Remove these edges



Example (cont.)

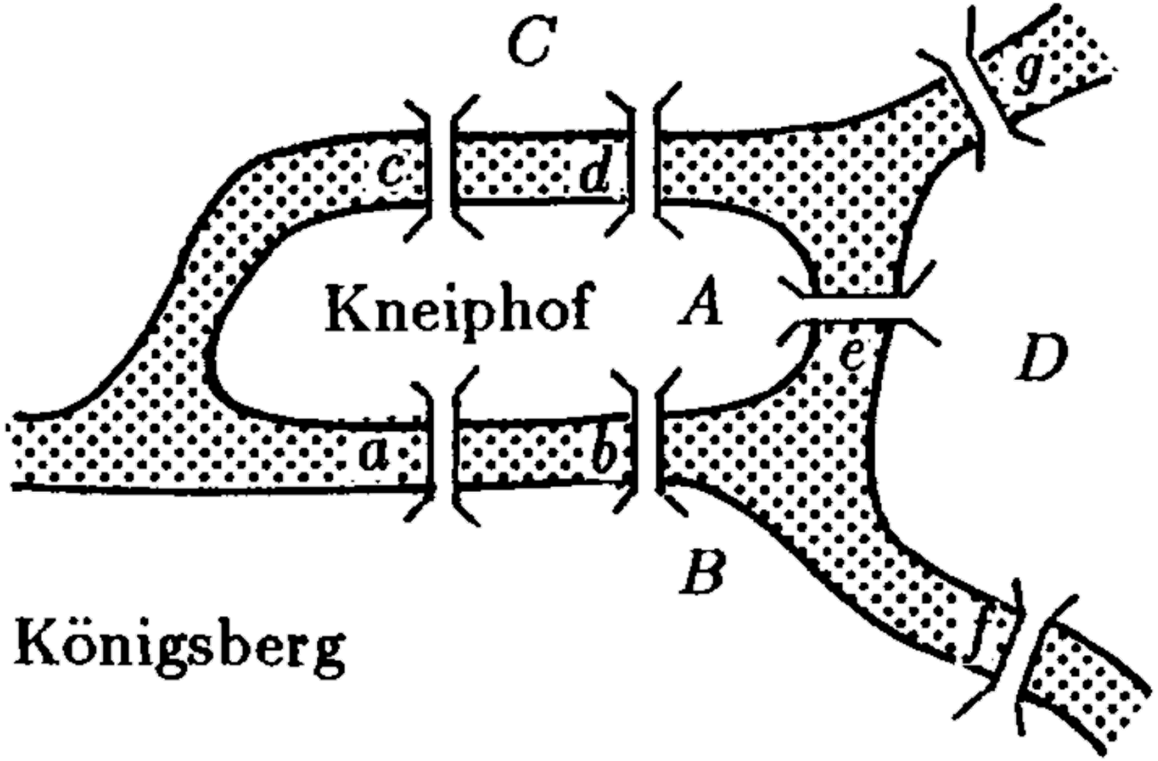
6. Expanding (1,2,5,8,2,4,3,1)
7. Remove these edges
8. First vertex incident with remaining edges is 4
9. Greedy algorithm yields (4,6,7,4,9,6,10,4)
10. Expanding (1,2,5,8,2,4,6,7,4,9,6,10,4,3,1)
11. Remove these edges
12. First vertex incident with remaining edges is 7
13. Greedy algorithm yields (7,9,11,7)
14. Expanding (1,2,5,8,2,4,6,7,9,11,7,4,9,6,10,4,3,1)

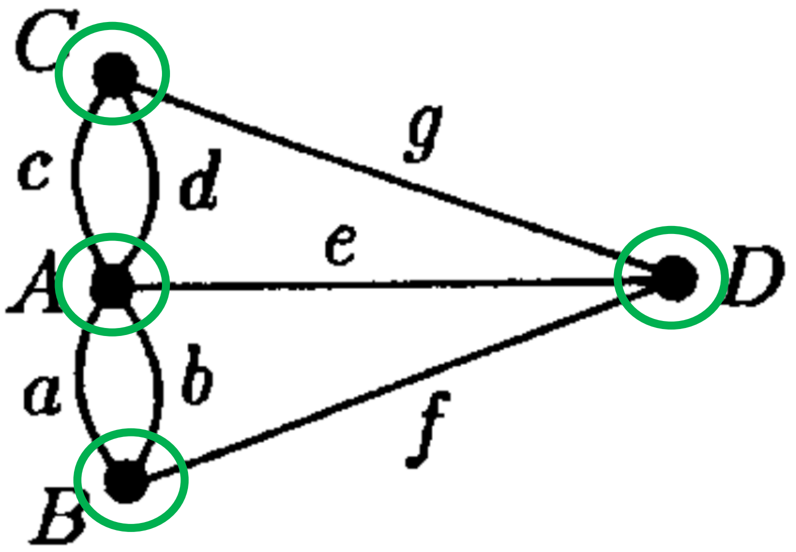




Eulerian circuit

- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree





TONCAS

- TONCAS: The obvious necessary condition is also sufficient
- Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree

Hamiltonian path/circuits

- A path P is Hamiltonian if $V(P) = V(G)$
  - Any graph contains a Hamiltonian path is called traceable
- A cycle C is called Hamiltonian if it spans all vertices of G
  - A graph is called Hamiltonian if it contains a Hamiltonian circuit
- In the mid-19th century, Sir William Rowan Hamilton tried to popularize the exercise of finding such a closed path in the graph of the dodecahedron

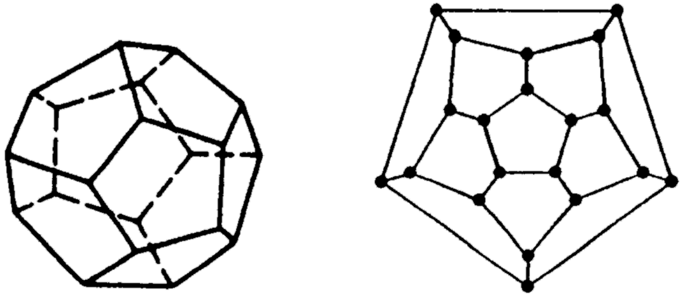
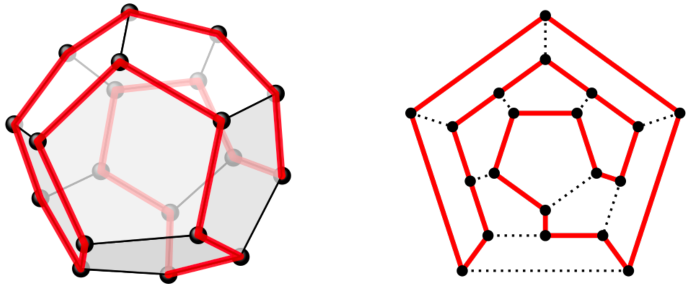


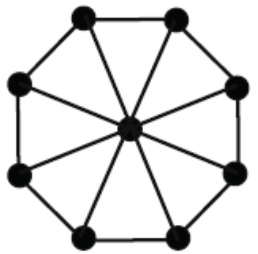
Figure 1.9



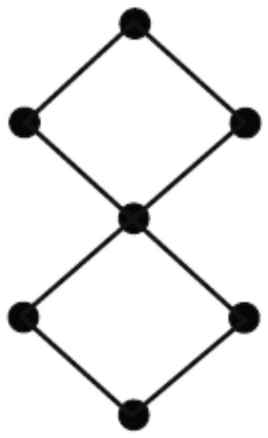
Degree parity is not a criterion

Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree

- Hamiltonian graphs
  - all even degrees C_{10}
  - all odd degrees K_{10}
  - a mixture G_1
- non-Hamiltonian graphs
  - all even G_2
  - all odd $K_{5,7}$
  - mixed P_9



G_1



G_2

Example

- The Petersen graph has a Hamiltonian path but no Hamiltonian cycle

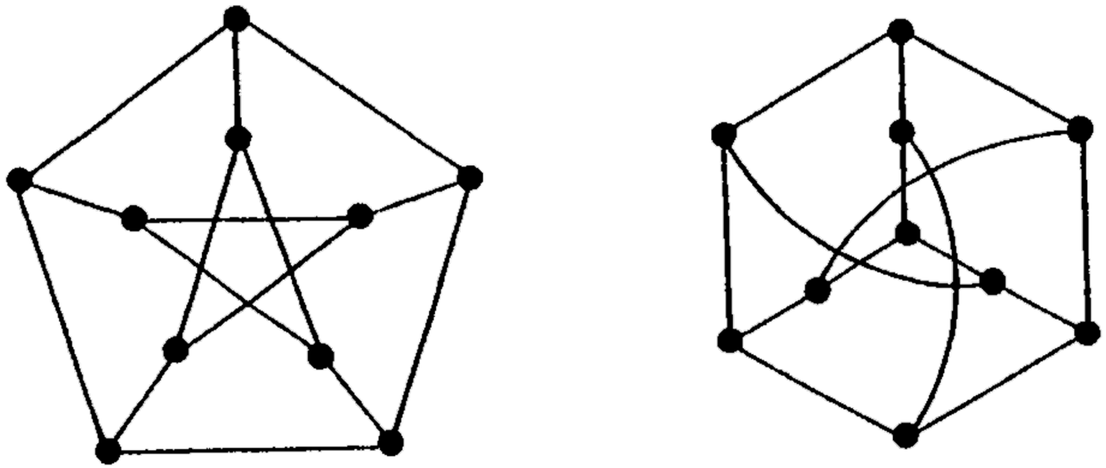
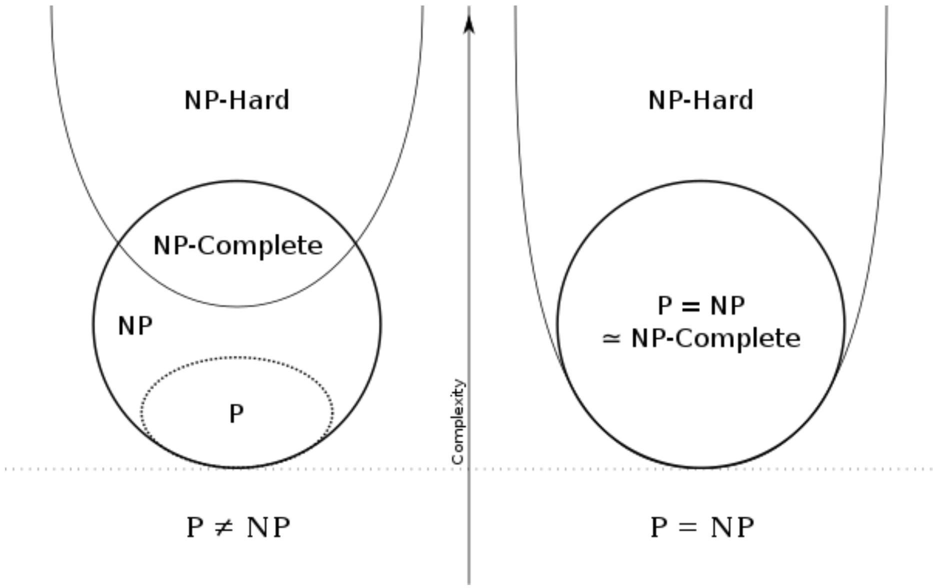


Figure 1.4

- Determining whether such paths and cycles exist in graphs is the Hamiltonian path problem, which is NP-complete

P, NP, NPC, NP-hard

- P The general class of questions for which some algorithm can provide an answer in polynomial time
- NP The class of questions for which an answer can be *verified* in polynomial time
- NP-Complete
  1. c is in NP
  2. Every problem in NP is reducible to c in polynomial time
- NP-hard
  - ~~c is in NP~~
  - Every problem in NP is reducible to c in polynomial time



Large minimal degree implies Hamiltonian

- Theorem (1.22, H, Dirac) Let G be a graph of order $n \geq 3$. If $\delta(G) \geq n/2$, then G is Hamiltonian

> Proposition (1.3.15, W) If $\delta(G) \geq \frac{n-1}{2}$, then G is connected
>
> (Ex16, S1.1.2, H) (1.3.16, W)
> If $\delta(G) \geq \frac{n-2}{2}$, then G need not be connected

- The bound is tight
  (Ex12b, S1.4.3, H) $G = K_{r,r+1}$ is not Hamiltonian
- The condition is not necessary
  - C_n is Hamiltonian but with small minimum (and even maximum) degree

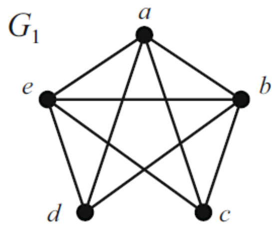
Generalized version

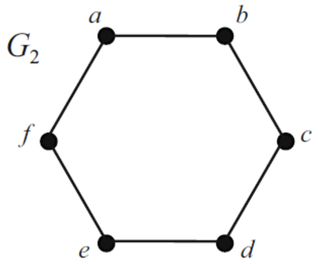
- Exercise (Theorem 1.23, H, Ore; Ex3, S1.4.3, H) Let G be a graph of order $n \geq 3$. If $\deg(x) + \deg(y) \geq n$ for all pairs of nonadjacent vertices x, y, then G is Hamiltonian

Theorem (1.22, H, Dirac) Let G be a graph of order $n \geq 3$. If $\delta(G) \geq n/2$, then G is Hamiltonian

Independence number & Hamiltonian

- A set of vertices in a graph is called independent if they are pairwise nonadjacent
- The independence number of a graph G, denoted as $\alpha(G)$, is the largest size of an independent set
- Example: $\alpha(G_1) = 2, \alpha(G_2) = 3$
- Theorem (1.24, H) Let G be a connected graph of order $n \geq 3$. If $\kappa(G) \geq \alpha(G)$, then G is Hamiltonian
- The result is tight: $\kappa(G) \geq \alpha(G) - 1$ is not enough
  - $K_{r,r+1}$: $\kappa = \mathrm{r}, \alpha = r + 1$
  - Peterson graph: $\kappa = 3, \alpha = 4$ (Ex4, S1.4.3, H)





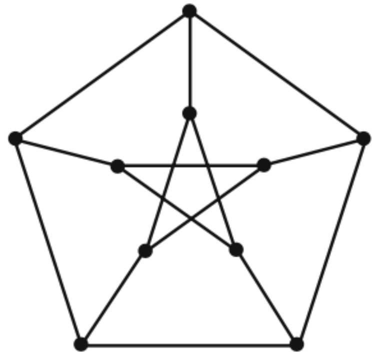
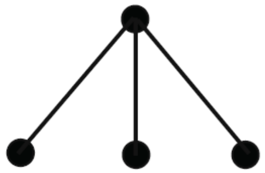
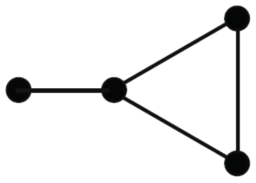


FIGURE 1.63. The Petersen Graph.

Pattern-free & Hamiltonian



$K_{1,3}$



Z_1

- G is H-free if G doesn't contain a copy of H as induced subgraph
- Theorem (1.25, H) If G is 2-connected and $\{K_{1,3}, Z_1\}$-free, then G is Hamiltonian

(Ex14, S1.1.2, H) $\kappa(G) \geq 2$ implies G has at least one cycle

- The condition 2-connectivity is necessary
- (Ex2, S1.4.3, H) If G is Hamiltonian, then G is 2-connected

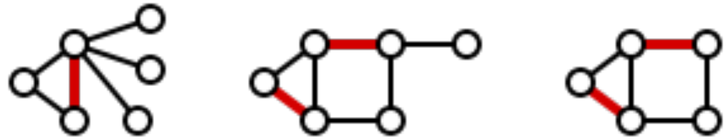
Matchings

Definitions

- A matching is a set of independent edges, in which no pair shares a vertex
- The vertices incident to the edges of a matching M are M-saturated; the others are M-unsaturated
- A perfect matching in a graph is a matching that saturates every vertex
- Example (3.1.2, W) The number of perfect matchings in $K_{n,n}$ is $n!$
- Example (3.1.3, W) The number of perfect matchings in K_{2n} is
$$f_n = (2n-1)(2n-3)\cdots 1 = (2n-1)!!$$

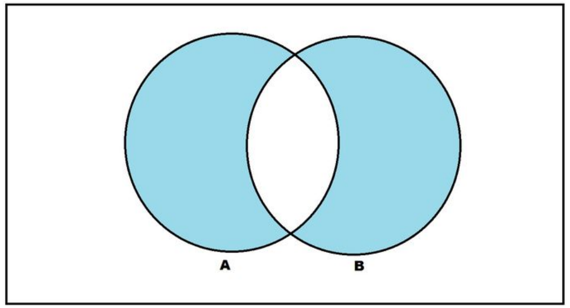
Maximal/maximum matchings 极大/最大

- A maximal matching in a graph is a matching that cannot be enlarged by adding an edge
- A maximum matching is a matching of maximum size among all matchings in the graph
- Example: P_3, P_5

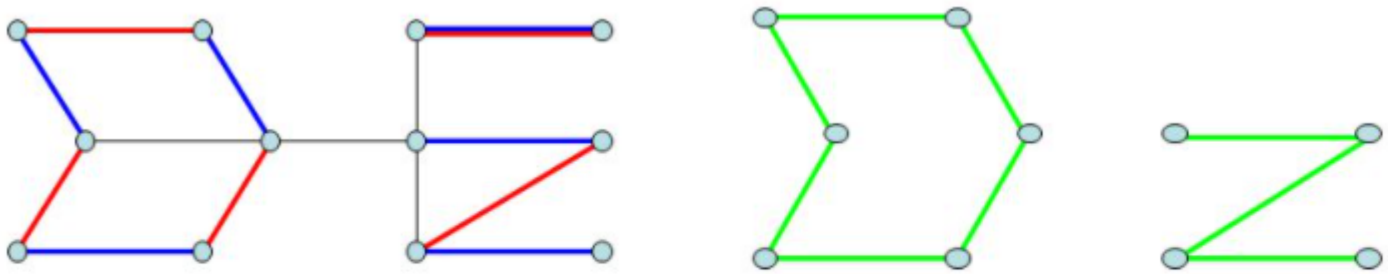


- Every maximum matching is maximal, but not every maximal matching is a maximum matching

Symmetric difference of matchings

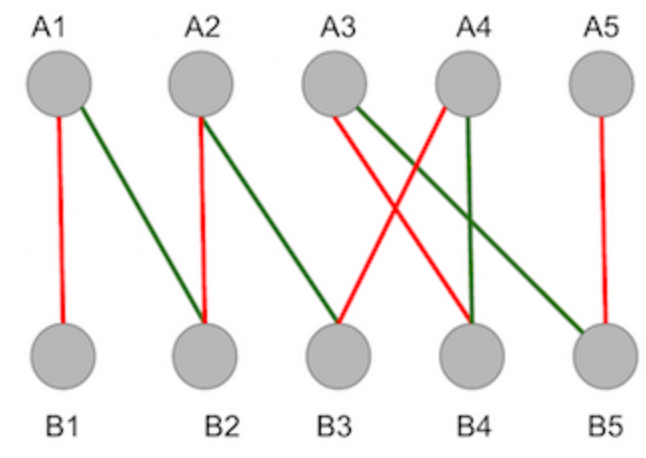


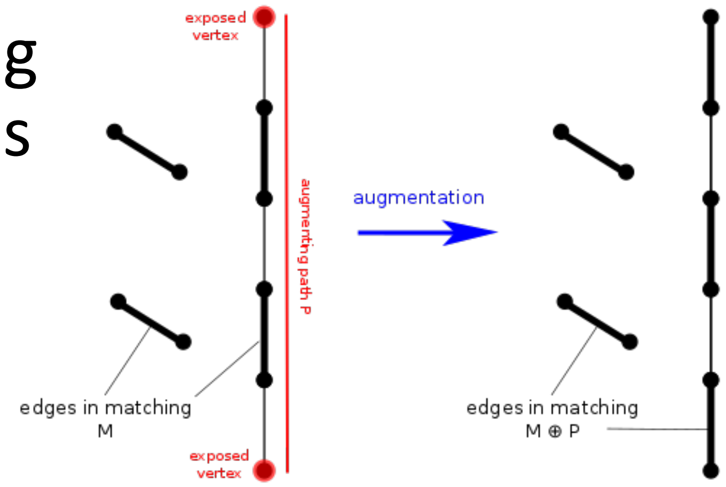
- The symmetric difference of M, M' is $M \Delta M' = (M - M') \cup (M' - M)$
- Lemma (3.1.9, W) Every component of the symmetric difference of two matchings is a path or an even cycle



Maximum matching and augmenting path

- Given a matching M, an M-alternating path is a path that alternates between edges in M and edges not in M
- An M-alternating path whose endpoints are M-unsaturated is an M-augmenting path
- Theorem (3.1.10, W; 1.50, H; Berge 1957) A matching M in a graph G is a maximum matching in $G \Leftrightarrow G$ has no M-augmenting path





Hall's theorem (TONCAS)

- Theorem (3.1.11, W; 1.51, H; 2.1.2, D; Hall 1935) Let G be a bipartite graph with partition X, Y.
  G contains a matching of $X \Leftrightarrow |N(S)| \geq |S|$ for all $S \subseteq X$

> Theorem (3.1.10, W; 1.50, H; Berge 1957) A matching M in a graph G is a maximum matching in $G \Leftrightarrow G$ has no M-augmenting path

- Corollary (3.1.13, W; 2.1.3, D) Every k-regular ($k > 0$) bipartite graph has a perfect matching

Vertex cover

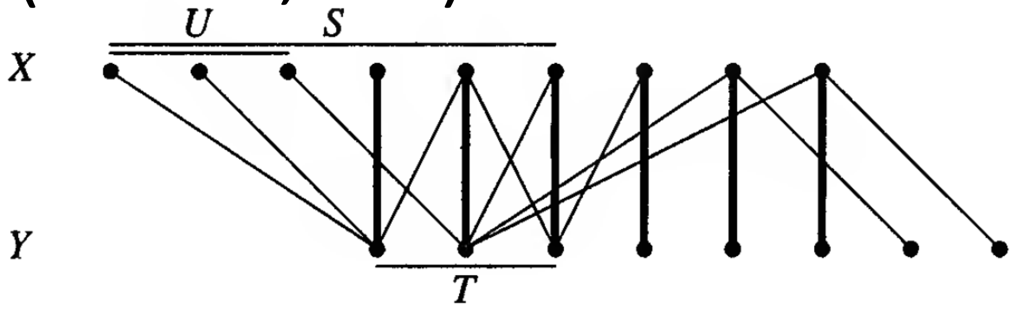
- A set $U \subseteq V$ is a (vertex) cover of E if every edge in G is incident with a vertex in U
- Example:
  - Art museum is a graph with hallways are edges and corners are nodes
  - A security camera at the corner will guard the paintings on the hallways
  - The minimum set to place the cameras?

König-Egeváry Theorem (Min-max theorem)

- Theorem (3.1.16, W; 1.53, H; 2.1.1, D; König 1931; Egeváry 1931) Let G be a bipartite graph. The maximum size of a matching in G is equal to the minimum size of a vertex cover of its edges

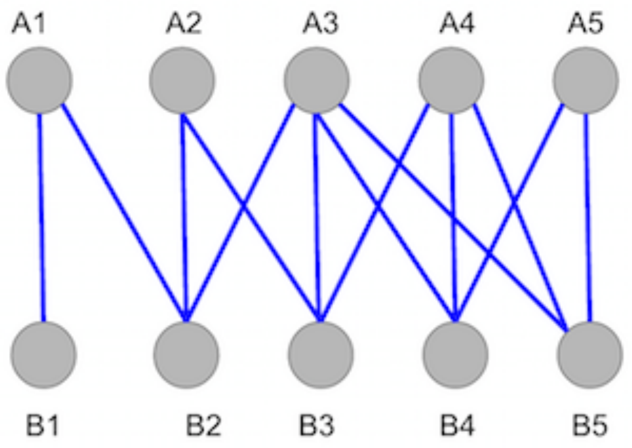
> Theorem (3.1.10, W; 1.50, H; Berge 1957) A matching M in a graph G is a maximum matching in $G \Leftrightarrow G$ has no M-augmenting path

Augmenting path algorithm (3.2.1, W)

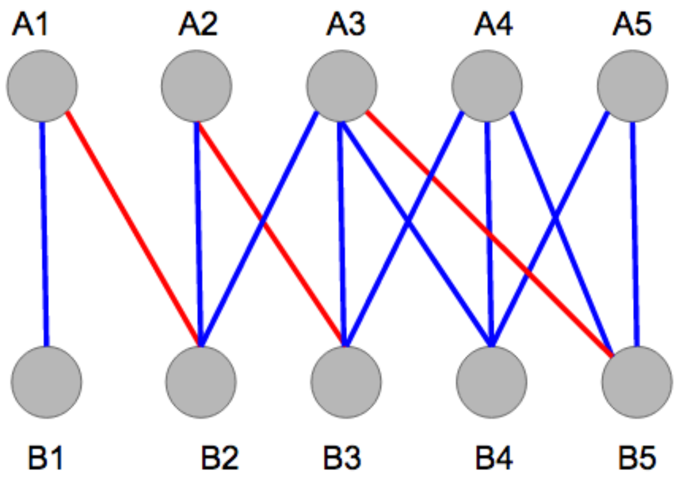


- **Input**: $G = B(X, Y)$, a matching M in G
  $U = \{M\text{–unsaturated vertices in } X\}$
- **Idea**: Explore M-alternating paths from U
  letting $S \subseteq X$ and $T \subseteq Y$ be the sets of vertices reached
- **Initialization**: $S = U, T = \emptyset$ and all vertices in S are unmarked
- **Iteration**:
  - If S has no unmarked vertex, stop and report $T \cup (X - S)$ as a minimum cover and M as a maximum matching
  - Otherwise, select an unmarked $x \in S$ to explore
    - Consider each $y \in N(x)$ such that $xy \notin M$
      - If y is unsaturated, terminate and report an M-augmenting path from U to y
      - Otherwise, $yw \in M$ for some w
        - include y in T (reached from x) and include w in S (reached from y)
    - After exploring all such edges incident to x, mark x and iterate.

Example



Red: A random matching



Theoretical guarantee for Augmenting path algorithm

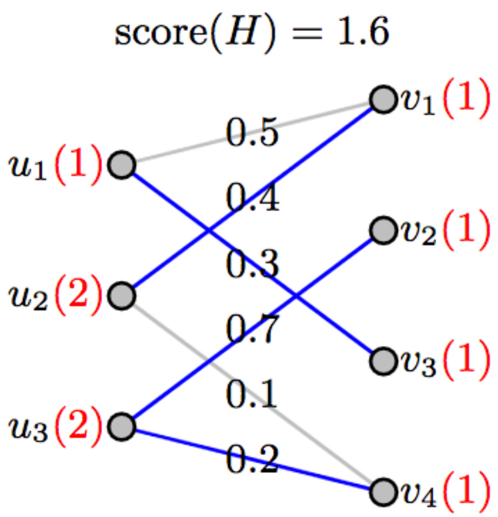
- Theorem (3.2.2, W) Repeatedly applying the Augmenting Path Algorithm to a bipartite graph produces a matching and a vertex cover of equal size

Weighted bipartite matching

- The maximum weighted matching problem is to seek a perfect matching M to maximize the total weight $w(M)$
- Bipartite graph
  - W.l.o.g. Assume the graph is $K_{n,n}$ with $w_{i,j} \geq 0$ for all $i, j \in [n]$
  - Optimization:

$$\max \sum_{i,j} a_{i,j} w_{i,j}$$
$$s.t.\ a_{i,1} + \cdots + a_{i,n} \leq 1 \text{ for any } i$$
$$a_{1,j} + \cdots + a_{n,j} \leq 1 \text{ for any } j$$
$$a_{i,j} \in \{0,1\}$$

  - Integer programming
  - General IP problems are NP-Complete



(Weighted) cover

- A (weighted) cover is a choice of labels $u_1, \dots, u_n$ and $v_1, \dots, v_n$ such that $u_i + v_j \geq w_{i,j}$ for all i, j
  - The cost $c(u, v)$ of a cover (u, v) is $\sum_i u_i + \sum_j v_j$
  - The minimum weighted cover problem is that of finding a cover of minimum cost
- Optimization problem

$$\min \sum_i u_i + \sum_j v_j$$
$$s.t.\, u_i + v_j \geq w_{i,j} \text{ for any } i, j$$
$$u_i, v_j \geq 0 \text{ for any } i, j$$

Duality

(IP)

$$\max \sum_{i,j} a_{i,j} w_{i,j}$$
$$s.t.\ a_{i,1} + \cdots + a_{i,n} \le 1 \text{ for any } i$$
$$a_{1,j} + \cdots + a_{n,j} \le 1 \text{ for any } j$$
$$a_{i,j} \in \{0,1\}$$

(Linear programming)

$$\max \sum_{i,j} a_{i,j} w_{i,j}$$
$$s.t.\ a_{i,1} + \cdots + a_{i,n} \le 1 \text{ for any } i$$
$$a_{1,j} + \cdots + a_{n,j} \le 1 \text{ for any } j$$
$$a_{i,j} \ge 0$$

(Dual)

$$\min \sum_{i} u_i + \sum_{j} v_j$$
$$s.t.\ u_i + v_j \ge w_{i,j} \text{ for any } i, j$$
$$u_i, v_j \ge 0$$

- Weak duality theorem
  - For each feasible solution a and (u, v)

$$\sum_{i,j} a_{i,j} w_{i,j} \le \sum_{i} u_i + \sum_{j} v_j$$

thus $\max \sum_{i,j} a_{i,j} w_{i,j} \le \min \sum_i u_i + \sum_j v_j$

Duality (cont.)

- Strong duality theorem
  - If one of the two problems has an optimal solution, so does the other one and that the bounds given by the weak duality theorem are tight

$$\max \sum_{i,j} a_{i,j} w_{i,j} = \min \sum_{i} u_i + \sum_{j} v_j$$

- Lemma (3.2.7, W) For a perfect matching M and cover (u, v) in a weighted bipartite graph G, $c(u, v) \geq w(M)$
  $c(u, v) = w(M) \Leftrightarrow$ M consists of edges $x_i y_j$ such that $u_i + v_j = w_{i,j}$
  In this case, M and (u, v) are optimal.

Equality subgraph

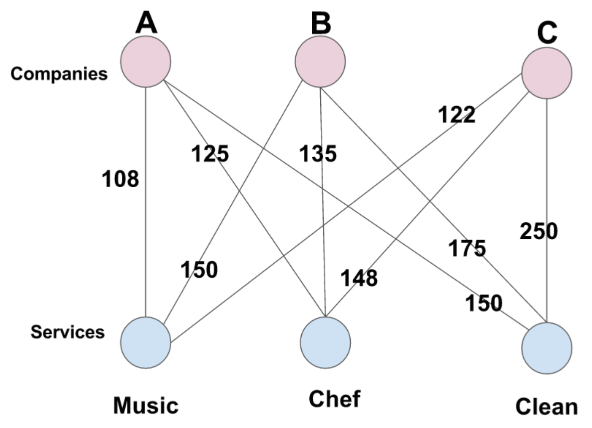
- The equality subgraph $G_{u,v}$ for a cover (u, v) is the spanning subgraph of $K_{n,n}$ having the edges $x_i y_j$ such that $u_i + v_j = w_{i,j}$
  - So if (u, v) is optimal, then M consistes the edges in $G_{u,v}$

Hungarian algorithm

- **Input**: Weighted $K_{n,n} = B(X,Y)$
- **Idea**: Iteratively adjusting the cover (u,v) until the equality subgraph $G_{u,v}$ has a perfect matching
- **Initialization**: Let (u,v) be a cover, such as $u_i = \max_j w_{i,j}$, $v_j = 0$

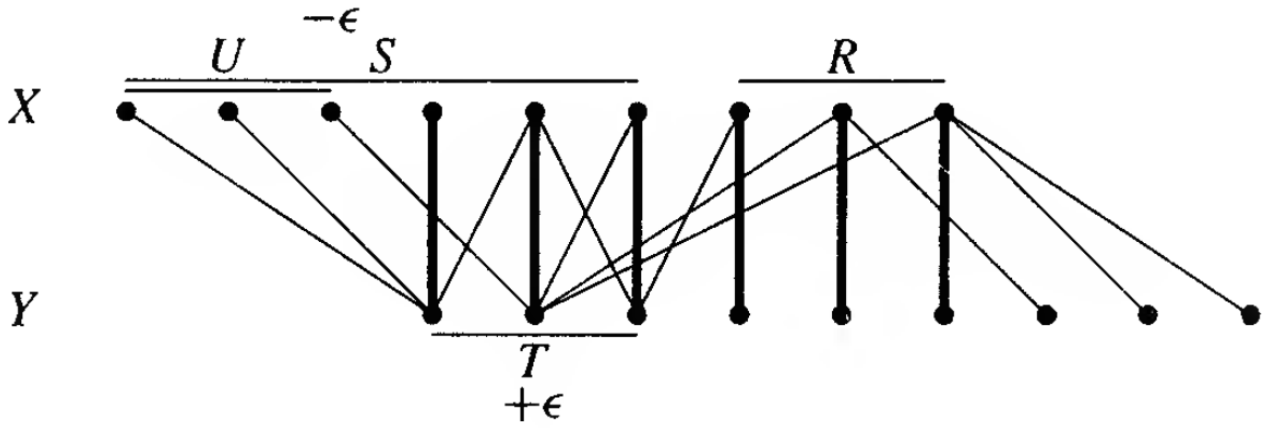
(Dual)

$$\min \sum_i u_i + \sum_j v_j$$
$$s.t.\, u_i + v_j \geq w_{i,j} \text{ for any } i,j$$
$$u_i, v_j \geq 0$$

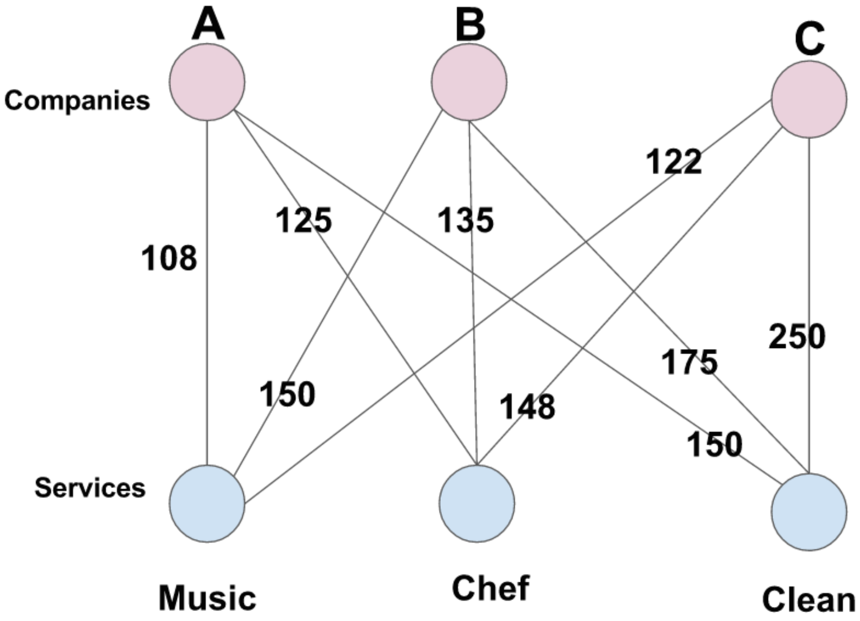


Hungarian algorithm (cont.)

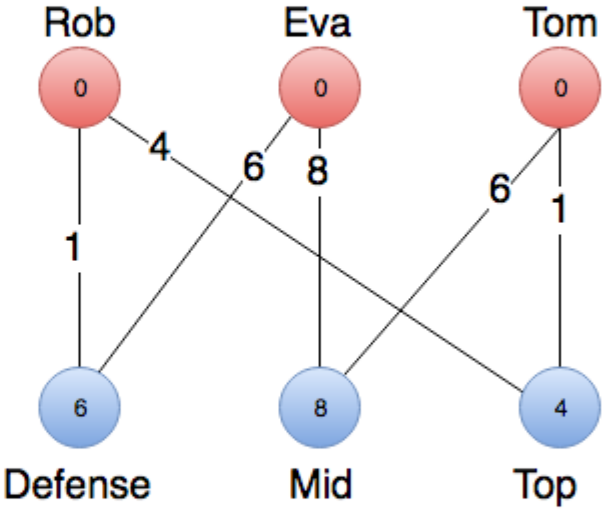
- **Iteration**: Find a maximum matching M in $G_{u,v}$
  - If M is a perfect matching, stop and report M as a maximum weight matching
  - Otherwise, let Q be a vertex cover of size $|M|$ in $G_{u,v}$
    - Let $R = X \cap Q, T = Y \cap Q$
    $$\epsilon = \min\{u_i + v_j - w_{i,j} : x_i \in X - R, y_j \in Y - T\}$$
    - Decrease u_i by ϵ for $x_i \in X - R$ and increase v_j by ϵ for $y_j \in T$
  - Form the new equality subgraph and repeat



Example



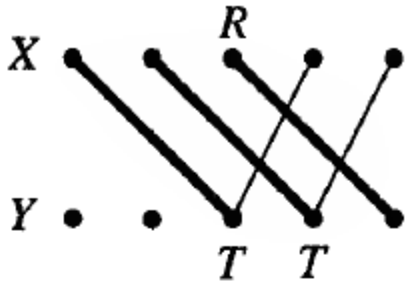
A
B
C
Companies
122
125
135
108
250
175
150
148
150
Services
Music
Chef
Clean



Rob
Eva
Tom
0
0
0
4
6
8
6
1
1
6
8
4
Defense
Mid
Top

Example 2

$$
\begin{pmatrix} 4 & 1 & 6 & 2 & 3 \\ 5 & 0 & 3 & 7 & 6 \\ 2 & 3 & 4 & 5 & 8 \\ 3 & 4 & 6 & 3 & 4 \\ 4 & 6 & 5 & 8 & 6 \end{pmatrix}
\rightarrow
\begin{array}{c}
\\ 6 \\ 7 \\ 8 \\ 6 \\ 8 \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \end{array} \\
\begin{pmatrix} 2 & 5 & \underline{0} & 4 & 3 \\ 2 & 7 & 4 & \underline{0} & 1 \\ 6 & 5 & 4 & 3 & \underline{0} \\ 3 & 2 & 0 & 3 & 2 \\ 4 & 2 & 3 & 0 & 2 \end{pmatrix} \\
\begin{array}{ccccc} & & T & T & \end{array}
\end{array}
\begin{array}{c} \\ \\ \\ R \\ \\ \\ \\ \end{array}
$$

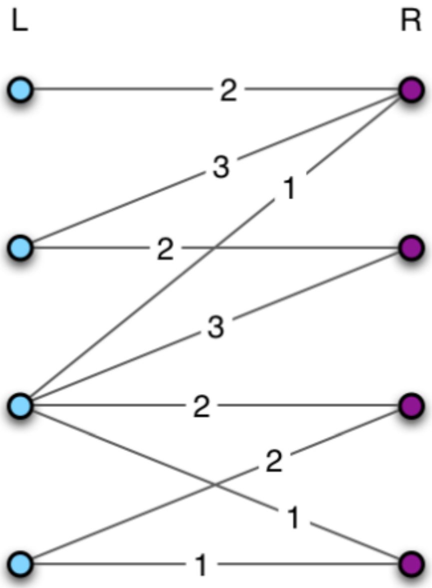


$$
\begin{array}{c}
\\ 5 \\ 6 \\ 8 \\ 5 \\ 7 \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 0 & 0 & 1 & 1 & 0 \end{array} \\
\begin{pmatrix} 1 & 4 & \underline{0} & 4 & 2 \\ 1 & 6 & 4 & \underline{0} & 0 \\ 6 & 5 & 5 & 4 & \underline{0} \\ 2 & 1 & 0 & 3 & 1 \\ 3 & 1 & 3 & 0 & 1 \end{pmatrix} \\
\begin{array}{ccccc} & & T & T & T \end{array}
\end{array}
$$

$$
\rightarrow
\begin{array}{c}
\\ 4 \\ 5 \\ 7 \\ 4 \\ 6
\end{array}
\begin{array}{c}
\begin{array}{ccccc} 0 & 0 & 2 & 2 & 1 \end{array} \\
\begin{pmatrix} 0 & 3 & \underline{0} & 4 & 2 \\ \underline{0} & 5 & 4 & 0 & 0 \\ 5 & 4 & 5 & 4 & \underline{0} \\ 1 & \underline{0} & 0 & 3 & 1 \\ 2 & 0 & 3 & \underline{0} & 1 \end{pmatrix}
\end{array}
$$

Optimal value is the same
But the solution is not unique

Example 3



Theoretical guarantee for Hungarian algorithm

- Theorem (3.2.11, W) The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover

Back to (unweighted) bipartite graph

- The weights are binary $0,1$
- Hungarian algorithm always maintain integer labels in the weighted cover, thus the solution will always be $0,1$
- The vertices receiving label 1 must cover the weight on the edges, thus cover all edges
- So the solution is a minimum vertex cover

Stable matching

- A family $(\leq_v)_{v \in V}$ of linear orderings $\leq_v$ on $E(v)$ is a set of preferences for G
- A matching M in G is stable if for any edge $e \in E \setminus M$, there exists an edge $f \in M$ such that e and f have a common vertex v with $e <_v f$
  - Unstable: There exists $xy \in E \setminus M$ but $xy', x'y \in M$ with $xy' <_x xy$
    $x'y <_y xy$

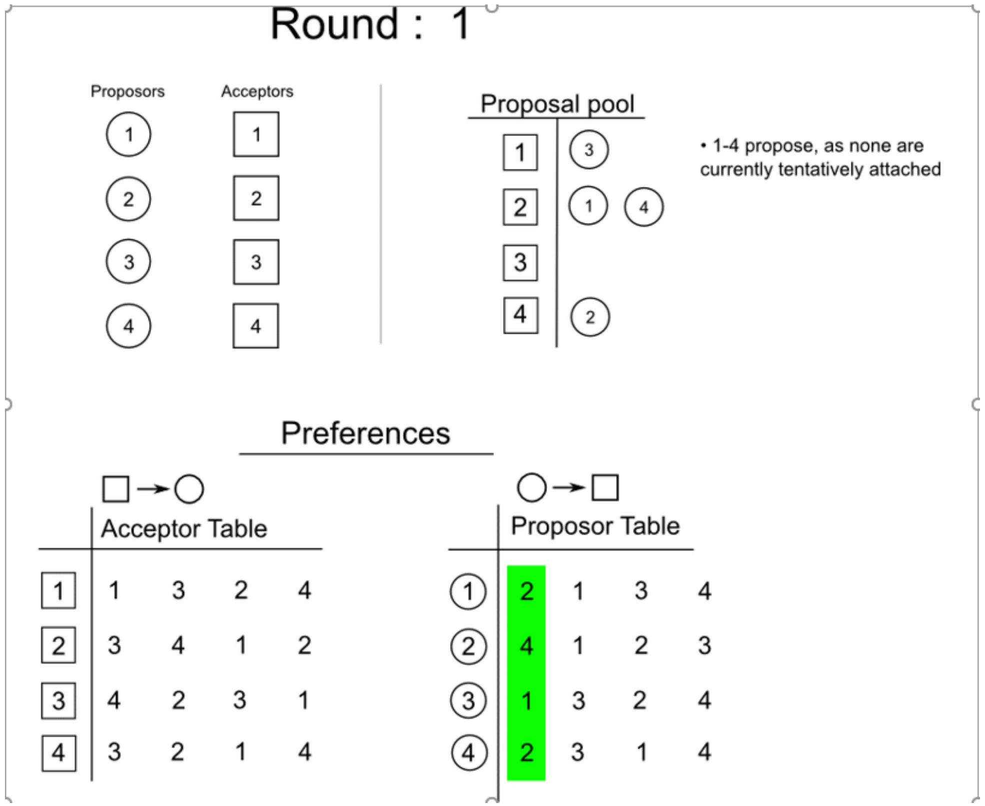
3.2.16. Example. Given men x, y, z, w, women a, b, c, d, and preferences listed below, the matching $\{xa, yb, zd, wc\}$ is a stable matching. ■

Men $\{x, y, z, w\}$	Women $\{a, b, c, d\}$
$x : a > b > c > d$	$a : z > x > y > w$
$y : a > c > b > d$	$b : y > w > x > z$
$z : c > d > a > b$	$c : w > x > y > z$
$w : c > b > a > d$	$d : x > y > z > w$

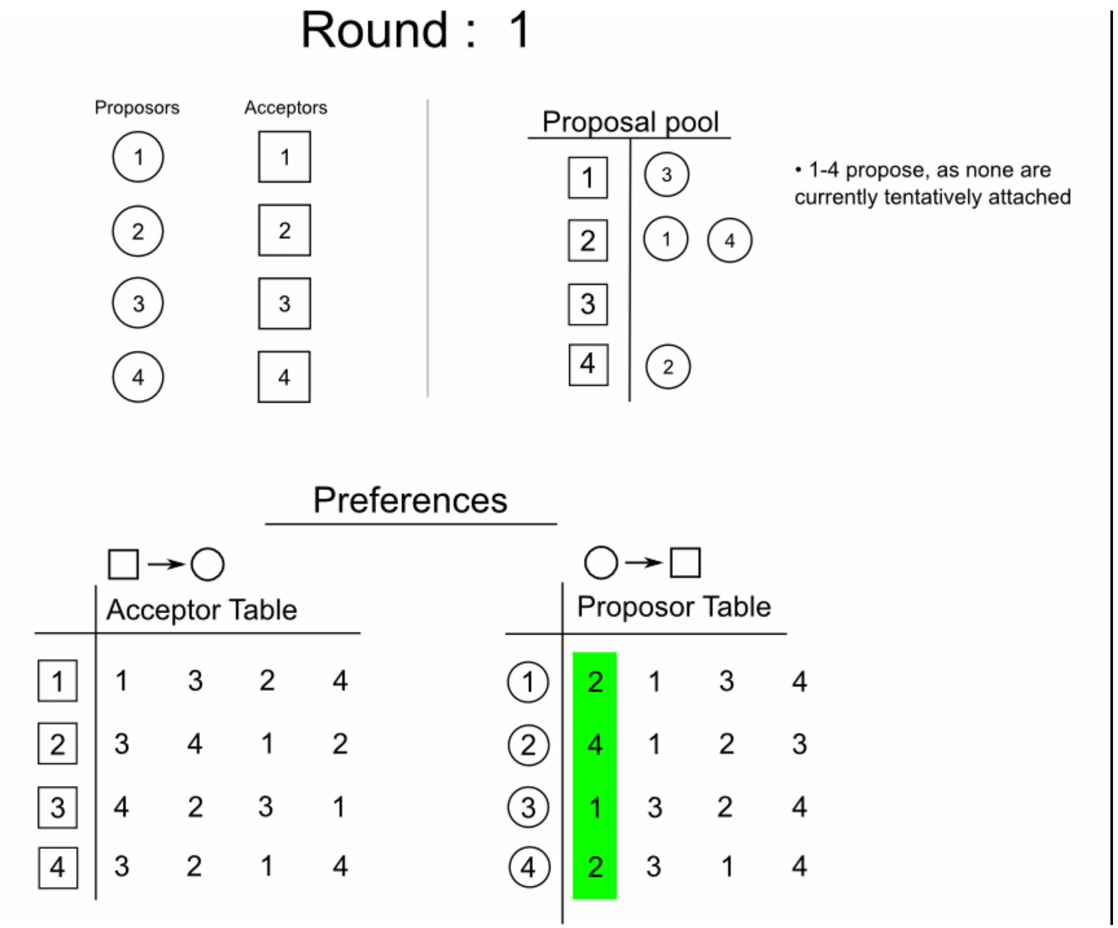
Gale-Shapley Proposal Algorithm

- **Input**: Preference rankings by each of n men and n women
- **Idea**: Produce a stable matching using proposals by maintaining information about who has proposed to whom and who has rejected whom
- **Iteration**: Each man proposes to the highest woman on his preference list who has not previously rejected him
  - If each woman receives exactly one proposal, stop and use the resulting matching
  - Otherwise, every woman receiving more than one proposal rejects all of them except the one that is highest on her preference list
  - Every woman receiving a proposal says “maybe” to the most attractive proposal received

Example


Round : 1
Proposors
Acceptors
1 2 3 4
1 2 3 4
Proposal pool
1: 3
2: 1 4
3:
4: 2
• 1-4 propose, as none are currently tentatively attached
Preferences
Acceptor Table
1: 1 3 2 4
2: 3 4 1 2
3: 4 2 3 1
4: 3 2 1 4
Proposor Table
1: 2 1 3 4
2: 4 1 2 3
3: 1 3 2 4
4: 2 3 1 4

Example (gif)


Round : 1
Proposors
Acceptors
Proposal pool
• 1-4 propose, as none are currently tentatively attached
Preferences
Acceptor Table
Proposor Table
1 1 3 2 4
2 3 4 1 2
3 4 2 3 1
4 3 2 1 4
1 2 1 3 4
2 4 1 2 3
3 1 3 2 4
4 2 3 1 4

Theoretical guarantee for the Proposal Algorithm

- Theorem (3.2.18, W, Gale-Shapley 1962) The Proposal Algorithm produces a stable matching
- Who proposes matters (jobs/candidates)
- When the algorithm runs with women proposing, every woman is as least as happy as when men do the proposing
    - And every man is at least as unhappy

3.2.16. Example. Given men x, y, z, w, women a, b, c, d, and preferences listed below, the matching $\{xa, yb, zd, wc\}$ is a stable matching. ■

Men $\{x, y, z, w\}$	Women $\{a, b, c, d\}$
$x : a > b > c > d$	$a : z > x > y > w$
$y : a > c > b > d$	$b : y > w > x > z$
$z : c > d > a > b$	$c : w > x > y > z$
$w : c > b > a > d$	$d : x > y > z > w$

Perfect matchings

- K_{2n}, C_{2n}, P_{2n} have perfect matchings
- Corollary (3.1.13, W; 2.1.3, D) Every k-regular ($k > 0$) bipartite graph has a perfect matching

- Theorem(1.58, H) If G is a graph of order $2n$ such that $\delta(G) \geq n$, then G has a perfect matching

Theorem (1.22, H, Dirac) Let G be a graph of order $n \geq 3$. If $\delta(G) \geq n/2$, then G is Hamiltonian

Tutte's Theorem (TONCAS)

- Let $q(G)$ be the number of connected components with odd order
- Theorem (1.59, H; 2.2.1, D; 3.3.3, W)
  Let G be a graph of order $n \geq 2$. G has a perfect matching $\Leftrightarrow q(G - S) \leq |S|$ for all $S \subseteq V$

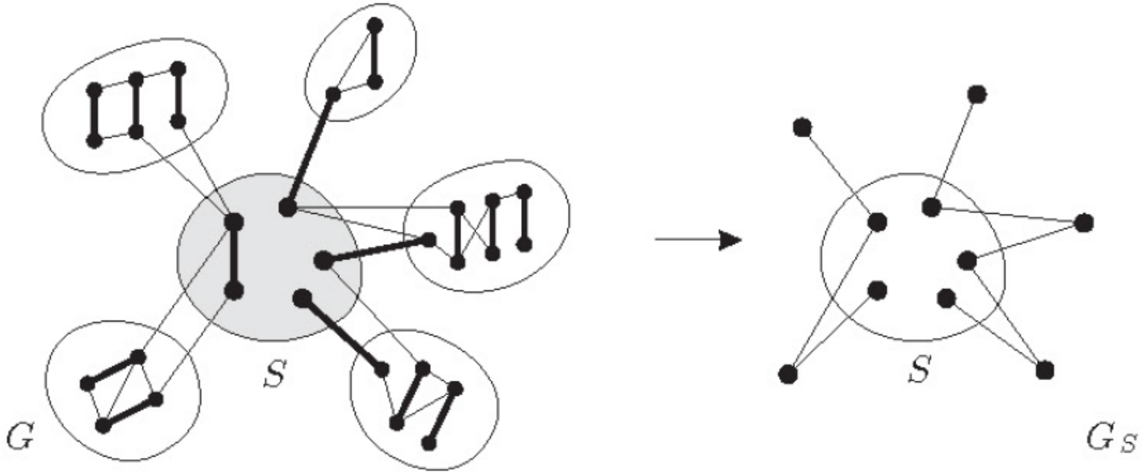


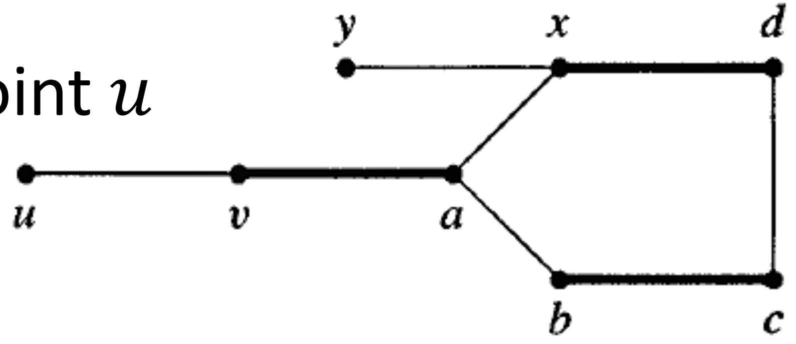
Fig. 2.2.1. Tutte's condition $q(G-S) \leqslant |S|$ for $q = 3$, and the contracted graph G_S from Theorem 2.2.3.

Petersen's Theorem

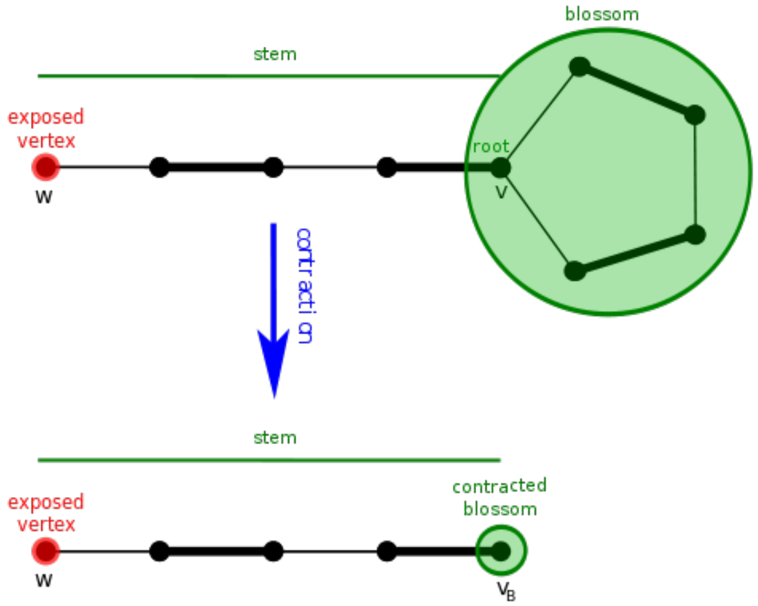
- Theorem (1.60, H; 2.2.2, D;3.3.8, W)
  Every bridgeless, 3-regular graph contains a perfect matching

Find augmenting paths in general graphs

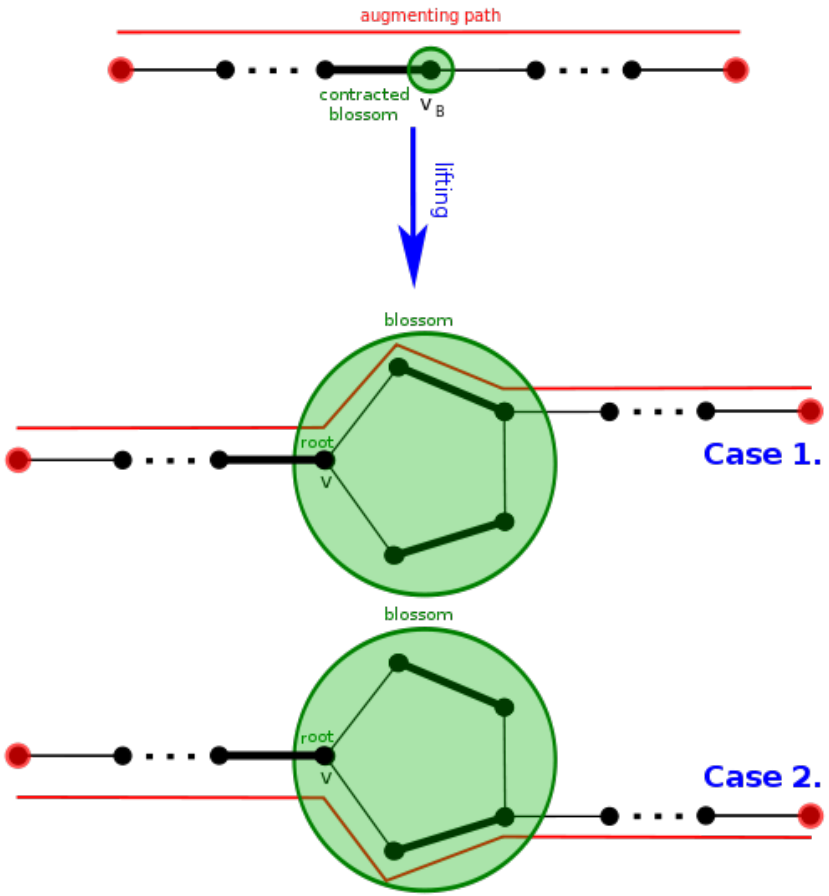
- Different from bipartite graphs
- Example: How to explore from M-unsaturated point u

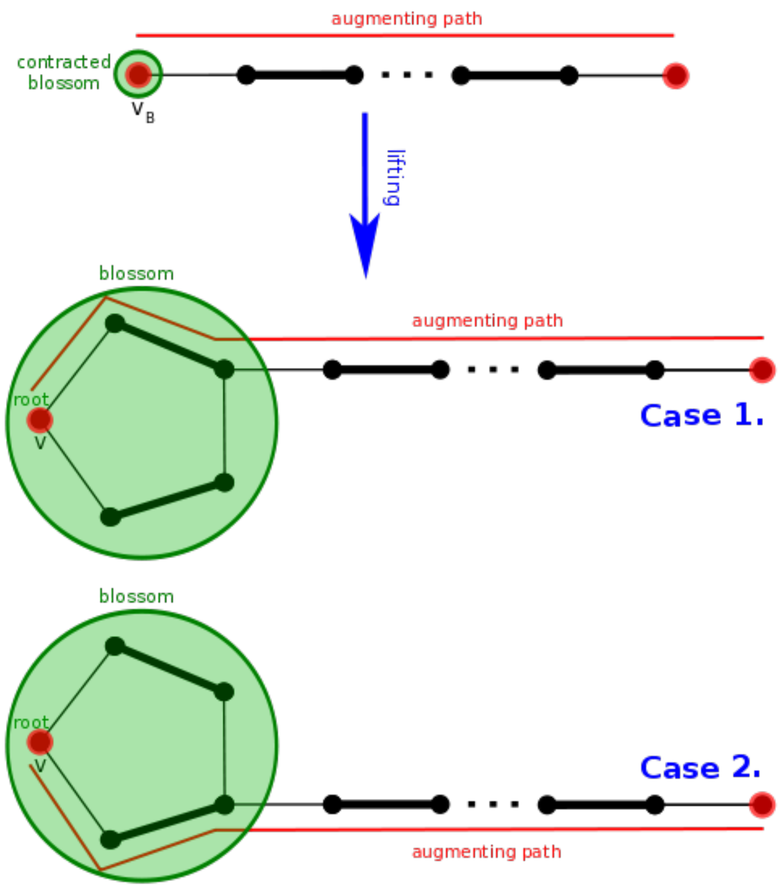


- Flower/stem/blossom



Lifting

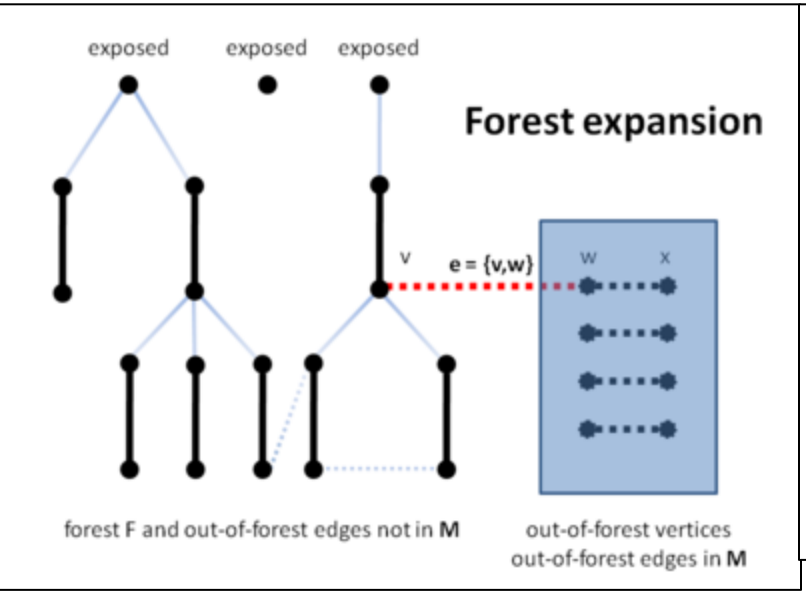

augmenting path
contracted blossom
v_B
lifting
blossom
root
v
Case 1.
blossom
root
v
Case 2.

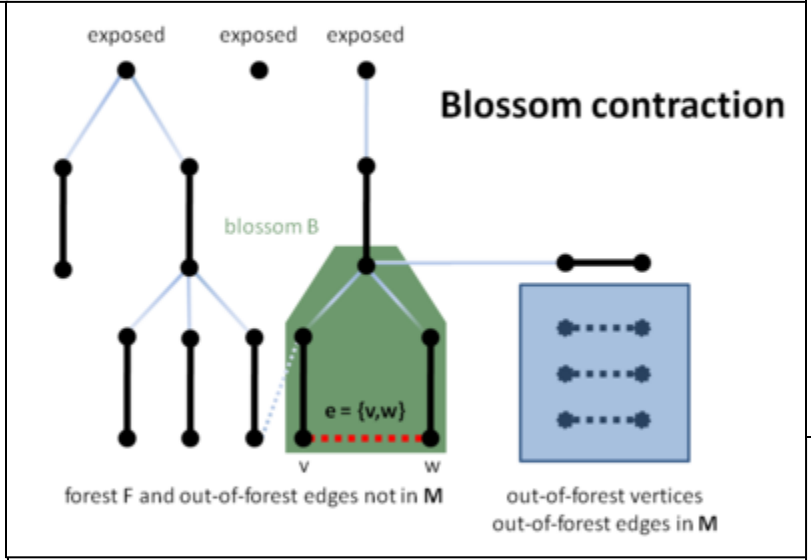

augmenting path
contracted blossom
v_B
lifting
blossom
augmenting path
root
v
Case 1.
blossom
root
v
Case 2.
augmenting path

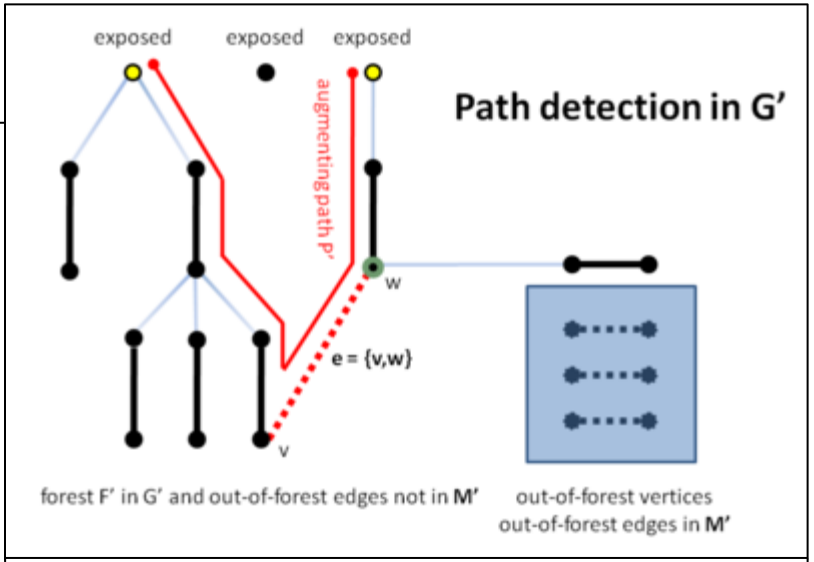
Edmonds' blossom algorithm (3.3.17, W)

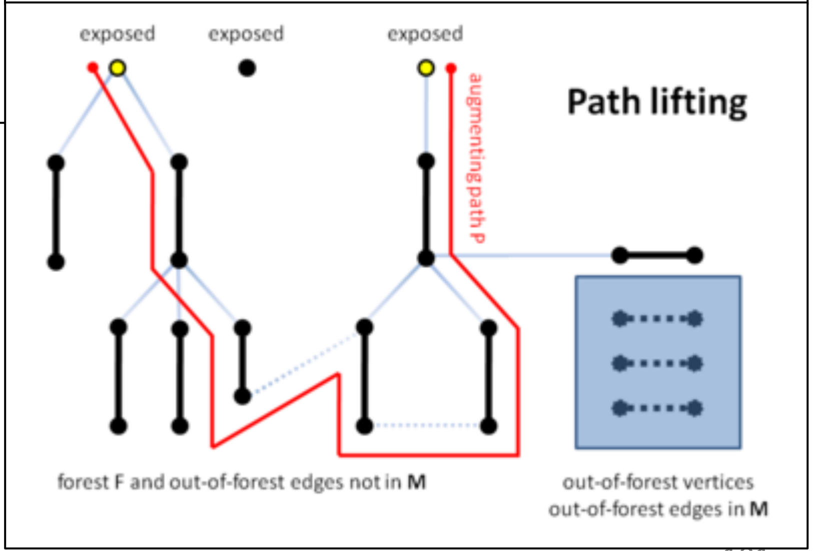
- **Input**: A graph G, a matching M in G, an M-unsaturated vertex u
- **Idea**: Explore M-alternating paths from u, recording for each vertex the vertex from which it was reached, and contracting blossoms when found
  - Maintain sets S and T analogous to those in Augmenting Path Algorithm, with S consisting of u and the vertices reached along saturated edges
  - Reaching an unsaturated vertex yields an augmentation.
- **Initialization**: $S = \{u\}$ and $T = \emptyset$
- **Iteration**: If S has no unmarked vertex, stop; there is no M-augmenting path from u
  - Otherwise, select an unmarked $v \in S$. To explore from v, successively consider each $y \in N(v)$ s.t. $y \notin T$
    - If y is unsaturated by M, then trace back from y (expanding blossoms as needed) to report an M-augmenting u, y-path
    - If $y \in S$, then a blossom has been found. Suspend the exploration of v and contract the blossom, replacing its vertices in S and T by a single new vertex in S. Continue the search from this vertex in the smaller graph.
    - Otherwise, y is matched to some w by M. Include y in T (reached from v), and include w in S (reached from y)
  - After exploring all such neighbors of v, mark v and iterate

Illustration

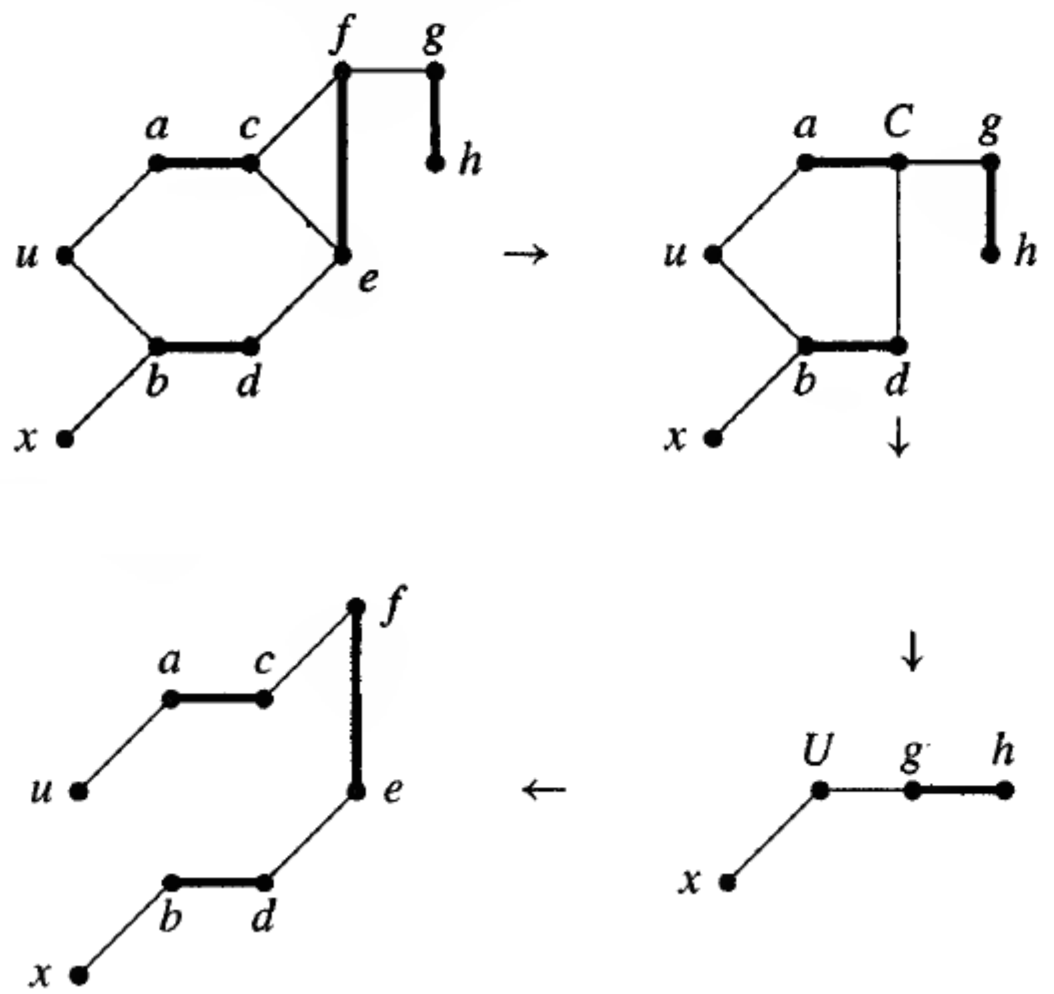








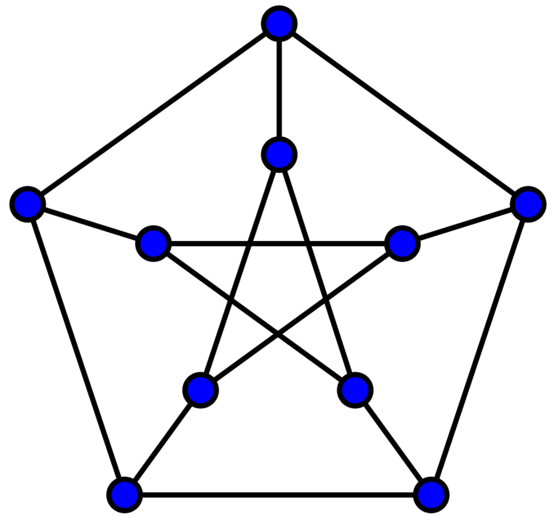
Example



TONCAS

- [Eulerian circuits] Theorem (1.2.26, W) A graph G is Eulerian $\Leftrightarrow$ it has at most one nontrivial component and its vertices all have even degree
- [Hall's theorem] (3.1.11, W; 1.51, H; 2.1.2, D; Hall 1935) Let G be a bipartite graph with partition $X, Y.$
  G contains a matching of $X \Leftrightarrow |N(S)| \geq |S|$ for all $S \subseteq X$
- [Tutte's Theorem](1.59, H; 2.2.1, D; 3.3.3, W)
  Let G be a graph of order $n \geq 2$. G has a perfect matching $\Leftrightarrow q(G - S) \leq |S|$ for all $S \subseteq V$

Peterson graph



- The Peterson graph is the unique 5-cage
  - cubic graph (every vertex has degree 3)
  - girth = 5
  - smallest graph satisfies the above properties
- $\kappa = 3, \alpha = 4$
- Radius=2, diameter=2
- The Petersen graph has a Hamiltonian path but no Hamiltonian cycle