Discrete Mathematics: Lecture 26

Paths and Isomorphism, Counting Paths, Euler Paths and Circuits

Xuming He
Associate Professor

School of Information Science and Technology
ShanghaiTech University

Spring Semester, 2022

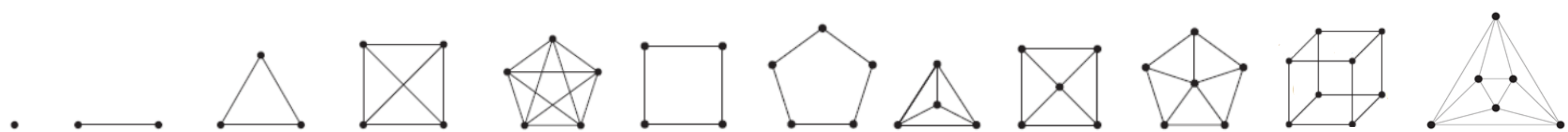
Notes by Prof. Liangfeng Zhang

Vertex Connectivity

DEFINITION: A connected undirected graph $G = (V, E)$ is said to be **nonseparable**不可分的 if G has no cut vertex.

DEFINITION: Let $G = (V, E)$ be a connected simple graph.

- **vertex cut**点割集: A subset $V' \subseteq V$ such that $G - V'$ is disconnected
- **vertex connectivity**点连通度 $\kappa(G)$: the minimum number of vertices whose removal disconnect G or results in K_1; equivalently,
  - if G is disconnected, $\kappa(G) = 0$; //additional definition
  - if $G = K_n, \kappa(G) = n - 1$ //K_n has no vertex cut
  - else, $\kappa(G)$ is the minimum size of a vertex cut of G



0 1 2 3 4 2 2 3 3 3 3 4

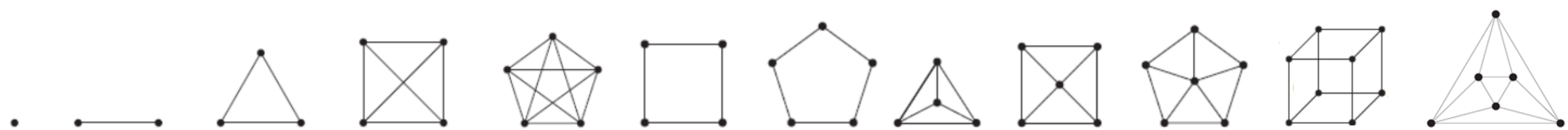
These graphs are all nonseparable

Edge Connectivity

DEFINITION: Let $G = (V, E)$ be a connected simple graph. $E' \subseteq E$ is an **edge cut**边割集 of G if $G - E'$ is disconnected.

DEFINITION: Let $G = (V, E)$ be a simple graph. The **edge connectivity**边连通度 $(\lambda(G))$ of G is defined as below:

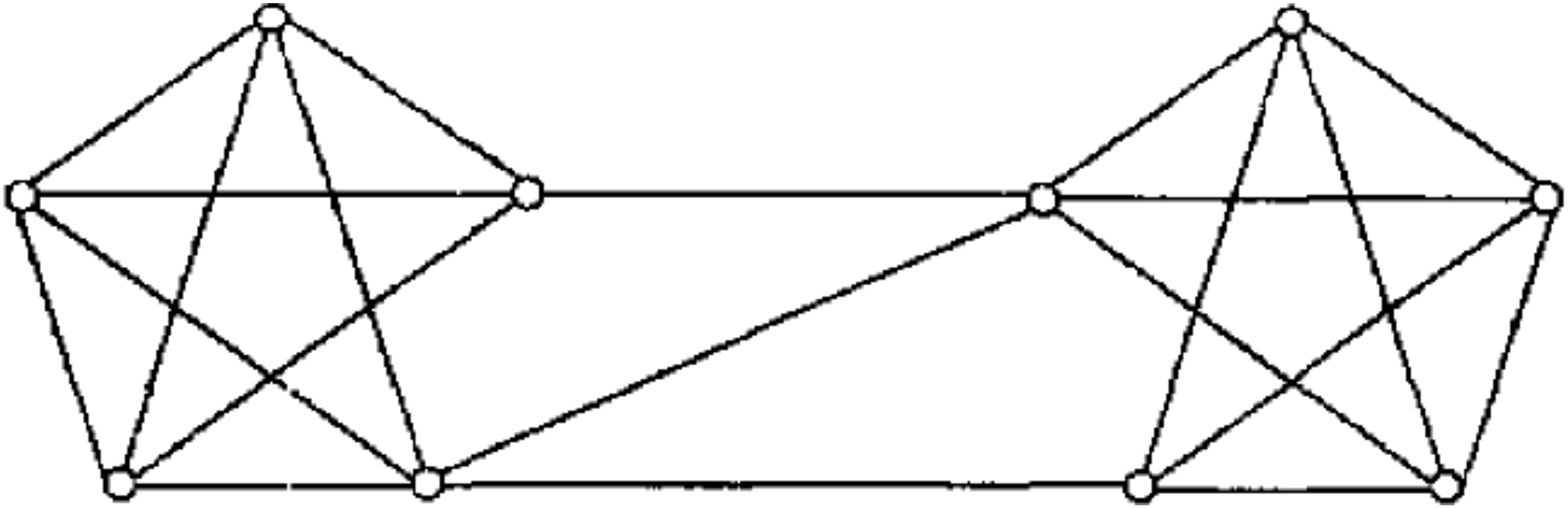
- G disconnected: $\lambda(G) = 0$
- G connected:
  - $|V| = 1$: $\lambda(G) = 0$
  - $|V| > 1$: $\lambda(G)$ is the minimum size of edge cuts of G.



0 1 2 3 4 2 2 3 3 3 3 4

Connectivity

THEOREM: Let $G = (V, E)$ be a simple graph. Then $\kappa(G) \leq \lambda(G) \leq \delta(G)$, where $\delta(G) = \min_{v \in V} \deg(v)$ is the least degree of $G's$ vertices.



- $\kappa(G) = 2$
- $\lambda(G) = 3$
- $\delta(G) = 4$

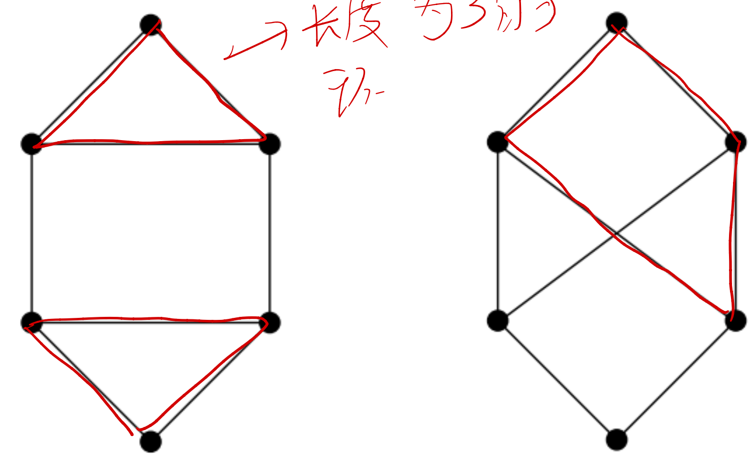
https://cp-algorithms.com/graph/edge_vertex_connectivity.html

http://www.math.caltech.edu/~2014-15/2term/ma006b/05%20connectivity%201.pdf

Paths and Isomorphism

Theorem

The existence of a simple circuit of length k, $k \geq 3$ is an isomorphism invariant for simple graphs.



长度为3的环

最小环长度为4

G_1 G_2

6 vertices, 8 edges

Degree sequence: 3, 3, 3, 3, 2, 2

Paths and Isomorphism*

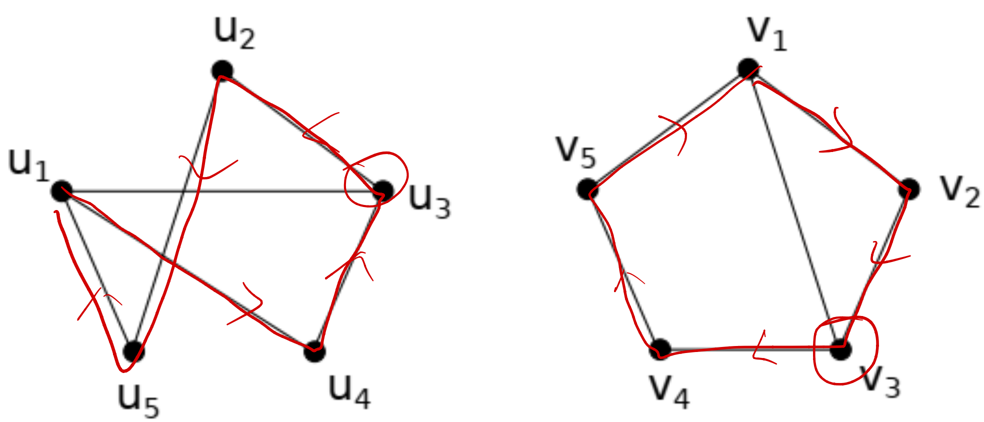
Theorem

The existence of a simple circuit of length k, $k \geq 3$ is an isomorphism invariant for simple graphs.

Proof: Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be isomorphic graphs: there is a bijective function $f : V_1 \to V_2$ respecting adjacency conditions.
Assume G_1 has a simple circuit of length k: $u_0, u_1, \ldots, u_k = u_0$, with $u_i \in V_1$ for $0 \leq i \leq k$. Let's denote $v_i = f(u_i)$, for $0 \leq i \leq k$.

$$(u_i, u_{i+1}) \in E_1 \Rightarrow (f(u_i), f(u_{i+1})) = (v_i, v_{i+1}) \in E_2, \text{ for } 0 \leq i \leq k-1.$$

So $v_0, \ldots, v_k$ is a path of length k in G_2.
It is a circuit because $v_k = f(u_k) = f(u_0) = v_0$.
It is simple: if not, at least one edge is traversed more than once, so it would mean that there exist $0 \leq i \neq j \leq k-1$ such that $(v_i, v_{i+1}) = (v_j, v_{j+1})$. But this implies $(u_i, u_{i+1}) = (u_j, u_{j+1})$ by bijectivity of f. This is impossible because $u_0, u_1, \ldots, u_k$ is simple.



G H

5 vertices, 6 edges
Degree sequence: 3, 3, 2, 2, 2
1 simple circuit of length 3,
1 simple circuit of length 4,
1 simple circuit of length 5.

Isomorphic graphs ?

If there is an iso $f : V_G \to V_H$, the simple circuit of length 5 u_1, u_4, u_3, u_2, u_5 must be sent to the simple circuit of length 5 in H, respecting the degrees of vertices. 度数必须一致

Check that $f(u_1) = v_1, f(u_4) = v_2, f(u_3) = v_3, f(u_2) = v_4, f(u_5) = v_5$ is an isomorphism by writing adjacency matrices.

Counting Paths Between Vertices

Theorem

Let G be a graph with adjacency matrix A with respect to the ordering of vertices $v_1, \ldots, v_n$. The number of different paths of length $r \geq 1$ from v_i to v_j equals the (i, j) entry of the matrix A^r.

Proof: By induction

- $r = 1$: the number of paths of length 1 from v_i to v_j is equal to the (i, j) entry of A by definition of A, as it corresponds to the number of edges from v_i to v_j.

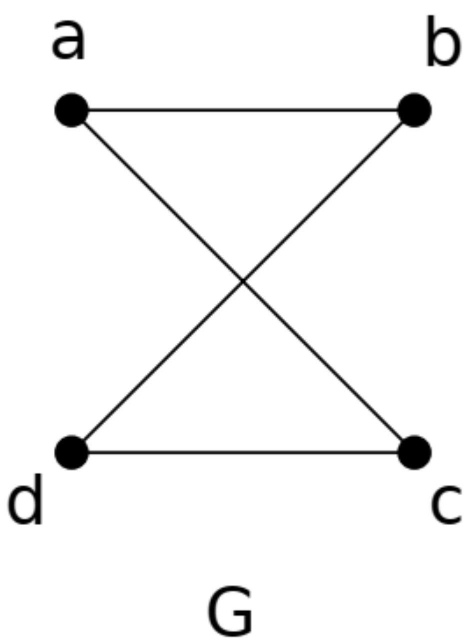
• Assume the (i, j) entry of the matrix A^r is the number of different paths of length r from v_i to v_j.
We can write $A^{r+1} = A^r A$
Let's denote $A^r = (b_{ij})_{1 \leq i,j \leq n}$, and $A = (a_{ij})_{1 \leq i,j \leq n}$. The (i, j) entry of A^{r+1} is given by:

$$\sum_{k=1}^{n} b_{ik} a_{kj} = b_{i1} a_{1j} + b_{i2} a_{2j} + \cdots + b_{in} a_{nj} \tag{1}$$

By hypothesis: b_{ik} equals the number of paths of length r from v_i to v_k.
"Path of length $r + 1$ from v_i to v_j = path of length r from v_i to any vertex v_k + an edge from v_k to v_j."
This is equal to the sum (1).

Example

How many paths of length four are there from a to d in the simple graph G



with ordering of vertices (a, b, c, d, e):

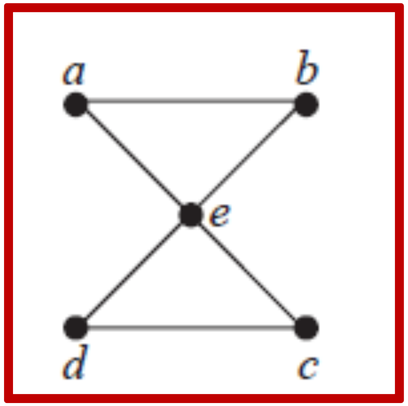
$$A_G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

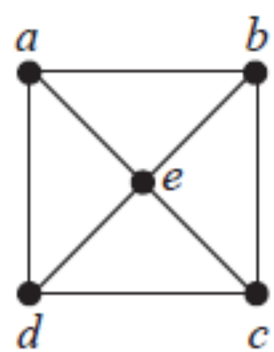
$$A_G^2 = \begin{pmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{pmatrix} \quad A_G^3 = \begin{pmatrix} 0 & 4 & 4 & 0 \\ 4 & 0 & 0 & 4 \\ 4 & 0 & 0 & 4 \\ 0 & 4 & 4 & 0 \end{pmatrix} \quad A_G^4 = \begin{pmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{pmatrix}$$

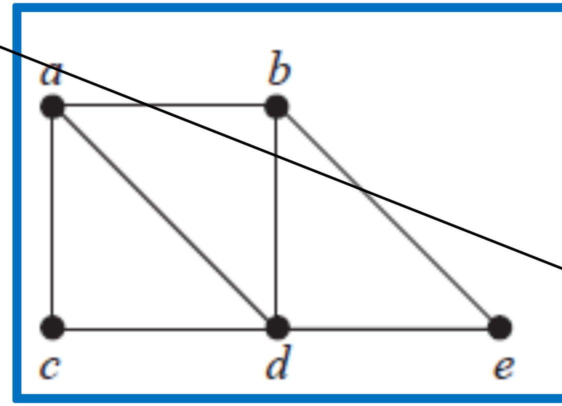
Euler Paths and Circuits

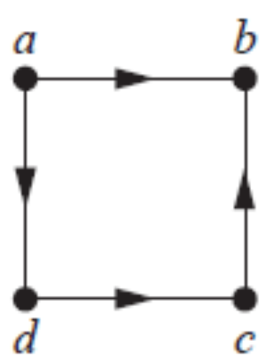
DEFINITION: Let $G = (V, E)$ be a graph. 一笔画，（点可重）

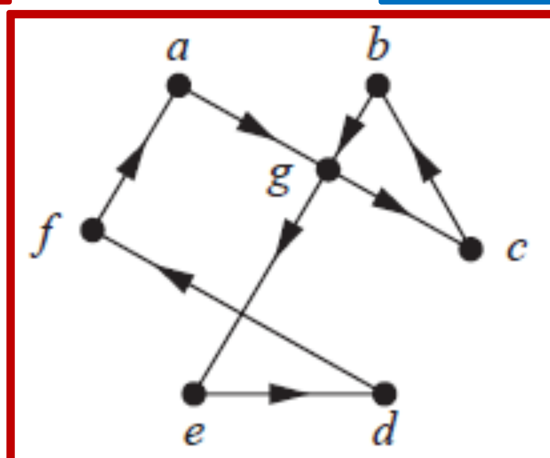
- **Euler Path**欧拉路径: a simple path that traverses every edge of G.
- **Euler Circuit**欧拉回路: a simple circuit that traverses every edge of G.

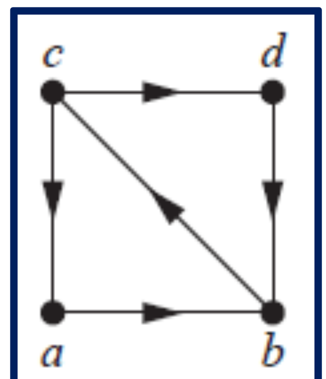












Remark: When G has multiple edges, these edges will be given different names and considered as different. This is implicit in the textbook.

Euler Circuits

THEOREM: Let $G = (V, E)$ be a connected multigraph of order ≥ 2.

Then G has an Euler circuit iff $2|\deg(x)$ for every $x \in V$.

- $\Rightarrow$: Let P: $\{x_0, x_1\}, \ldots, \{x_{i-1}, x_i\}, \ldots, \{x_{n-1}, x_n\}$ be an Euler circuit, $x_0 = x_n$
  - Every occurrence of x_i in P contributes 2 to $\deg(x_i)$
    - Every vertex x_i has an even degree
- $\Leftarrow$: Let P: $\{x_0, x_1\}, \ldots, \{x_{n-1}, x_n\}$ be a longest simple path in G.

  诱
  - Let $H = G[P]$, the subgraph of G induced by all edges in P
    - If $x_n \neq x_0$, then $\deg_H(x_n)$ is odd and so P cannot be longest.
      - $x_n = x_0$, P is a simple circuit, and $2|\deg_H(x_i)$ for all i.
    - If $\exists i \in \{0,1, \ldots, n-1\}$ such that $\deg_H(x_i) < \deg_G(x_i)$,
      - then $\exists y \in V$ such that $\{x_i, y\} \notin P$
        - $y, x_i, x_{i+1}, \ldots, x_n, x_1, \ldots, x_{i-1}, x_i$ is longer than P
    - Hence, $\deg_H(x_i) = \deg_G(x_i)$ for all $i \in \{0,1, \ldots, n-1\}$.
      - $V = \{x_0, x_1, \ldots, x_{n-1}\}$ and $H = G$.
        - P is an Euler circuit.

Remark: H contains all vertices of G. Otherwise, P can be extended.

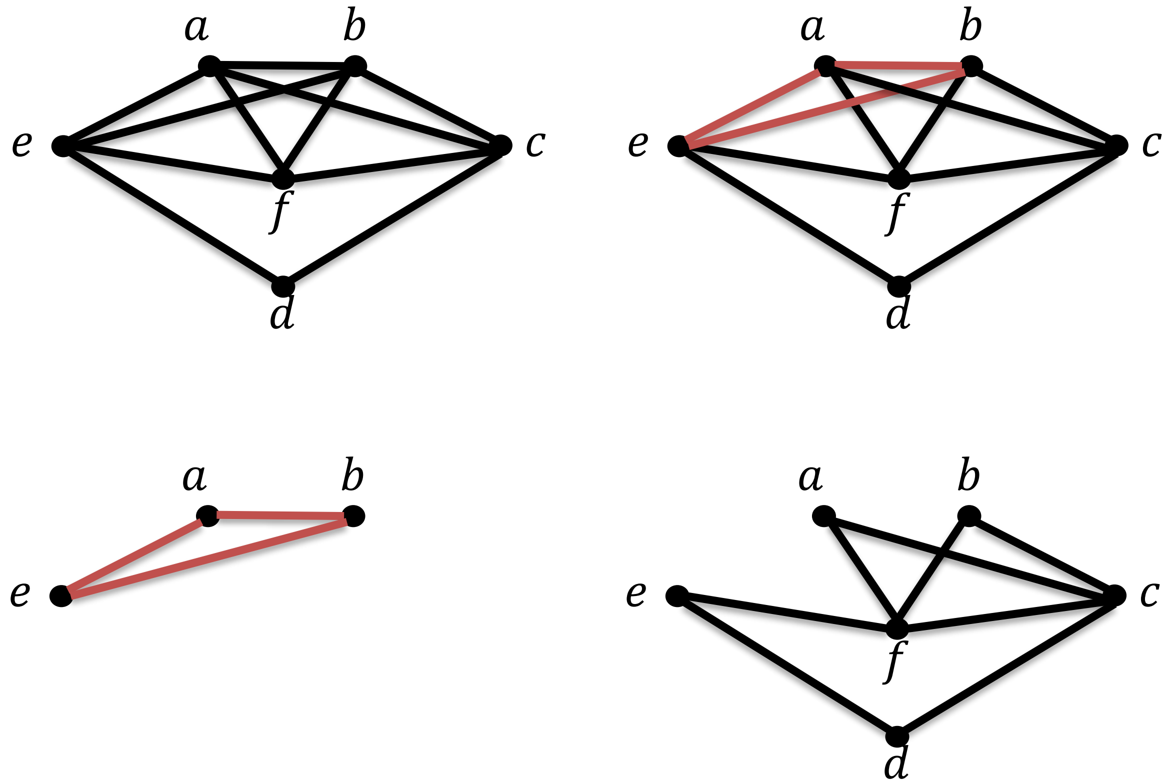
Construction

希尔霍尔策

ALGORITHM (Hierholzer):

- **Input**: $G = (V, E)$, a connected multigraph, $2 \mid \deg(x), \forall x \in V$
- **Output**: an Euler circuit
  - **circuit** $:=$ a circuit in G
  - $H :=$ $G -$ **circuit** $-$ isolated vertices
  - while H has edges do
    - **subcircuit** $:=$ a circuit in H that intersects **circuit**
    - $H :=$ $H -$ **subcircuit** $-$ isolated vertices
    - **circuit** $:=$ **circuit** $\cup$ **subcircuit**
  - return **circuit**

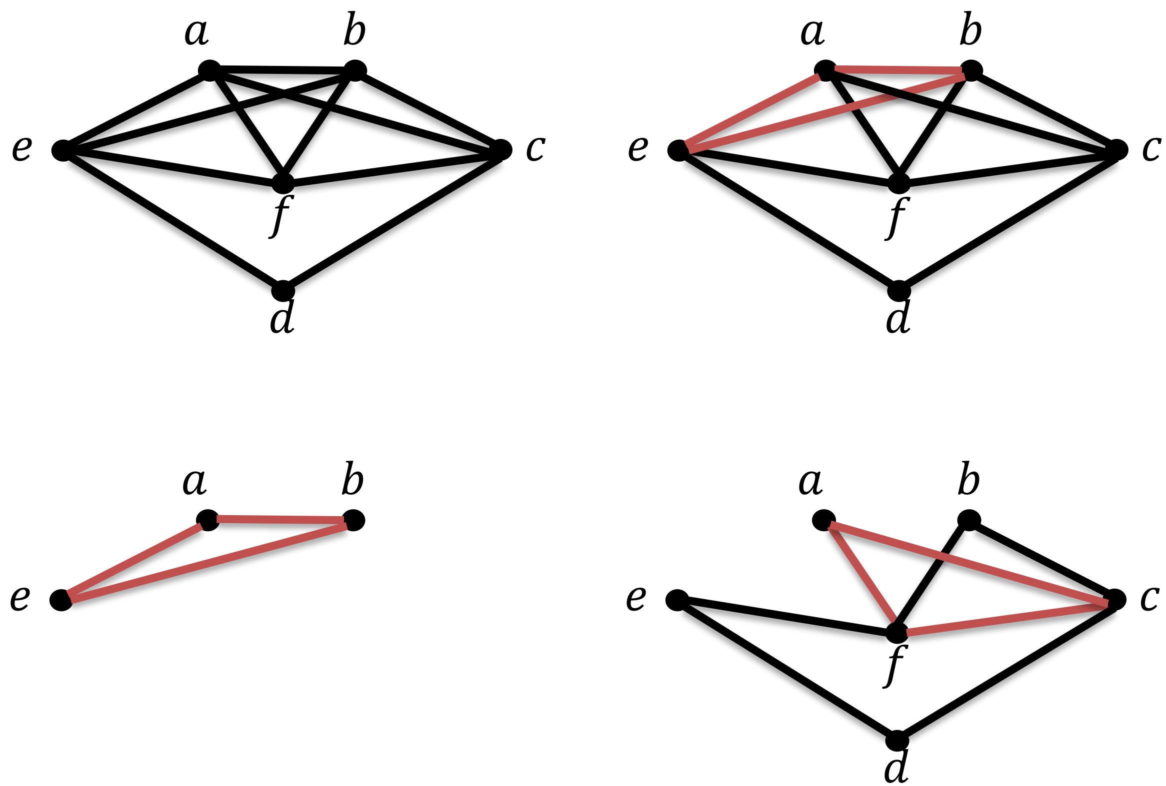
Example



circuit $= a, b, e, a$

H

Example

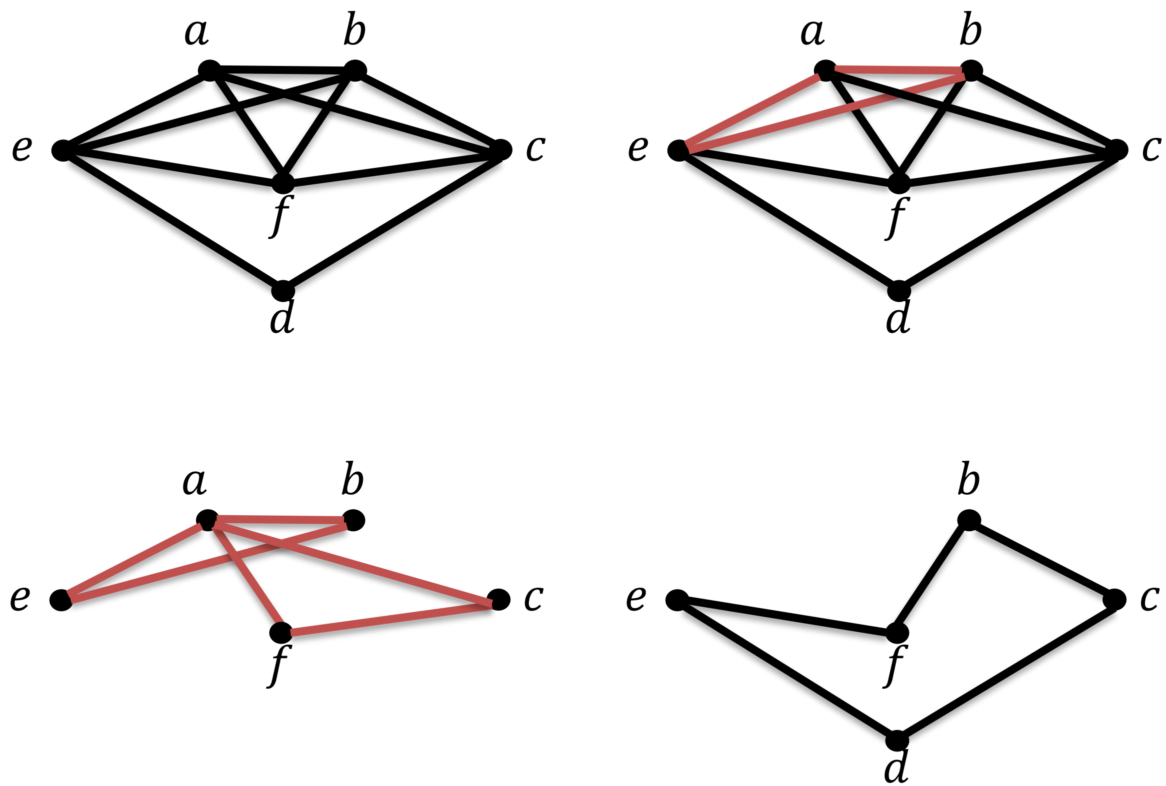


circuit $= a, b, e, a$

H

subcircuit $= a, c, f, a$

Example

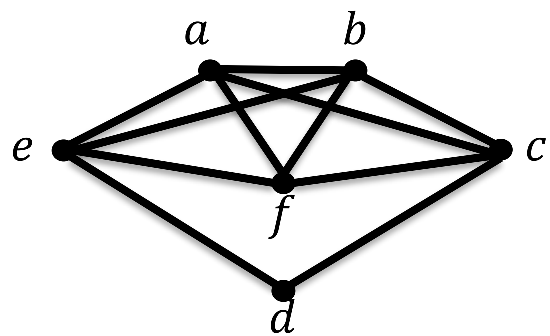


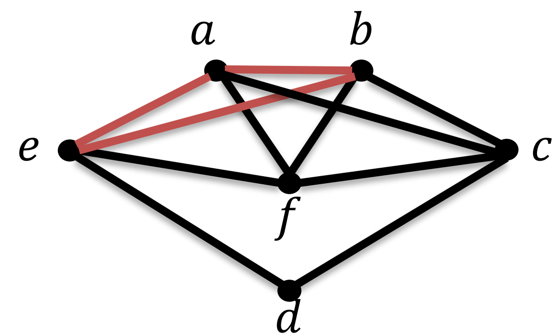
circuit $= a, b, e, a$

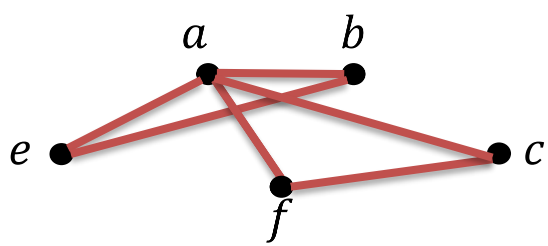
subcircuit $= a, c, f, a$

H

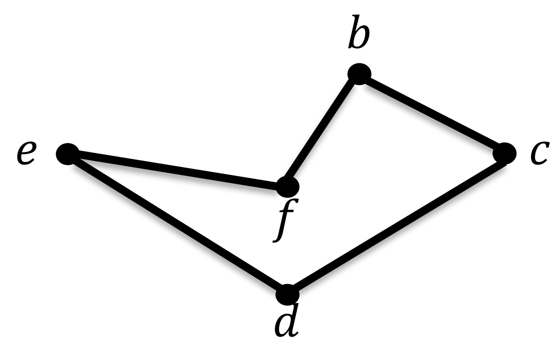
Example





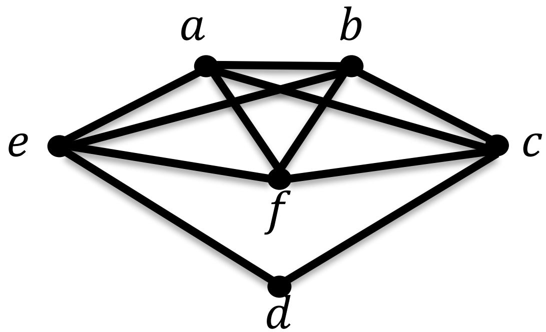


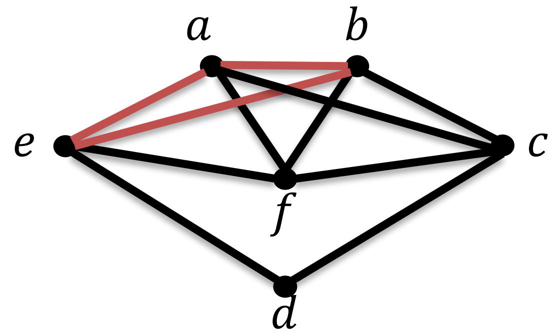
circuit $= a, b, e, a, c, f, a$

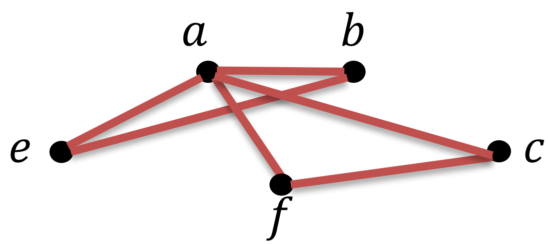


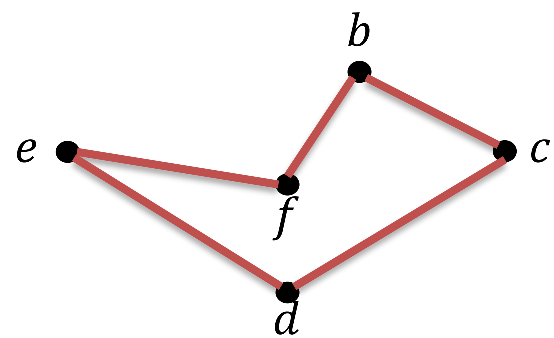
H

Example







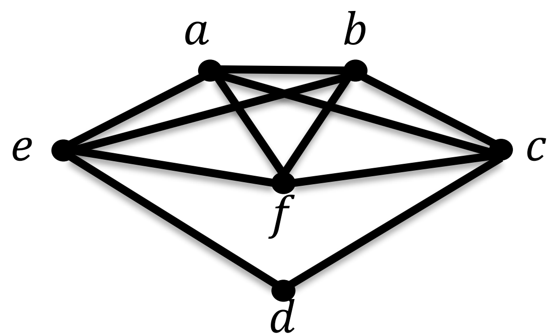


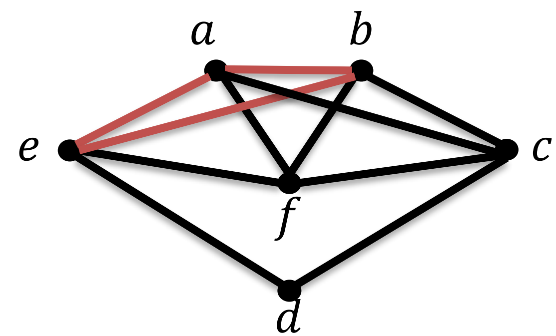
circuit $= a, b, e, a, c, f, a$

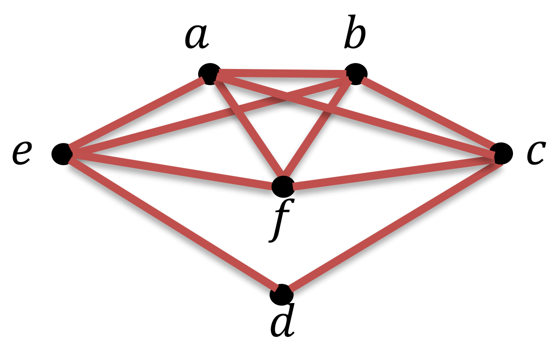
H

subcircuit $= c, d, e, f, b, c$

Example





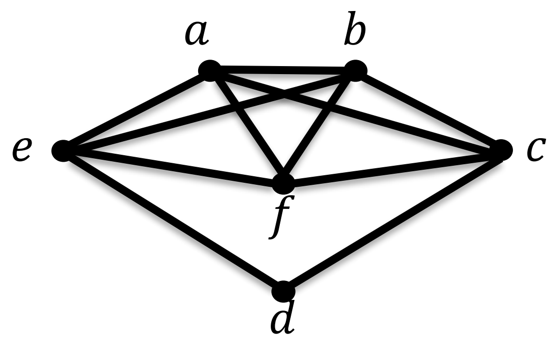


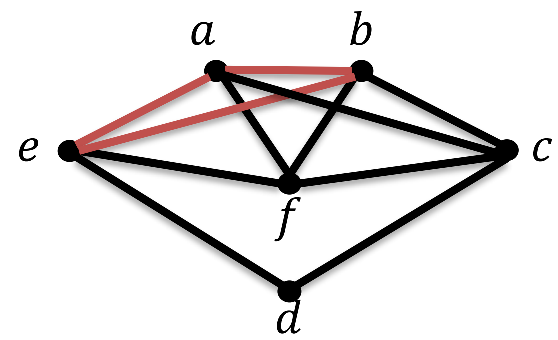
circuit $= a, b, e, a, c, f, a$

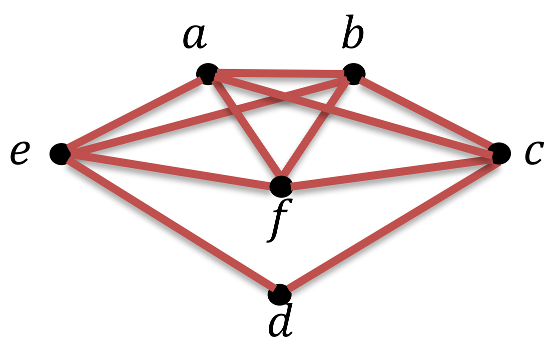
H

subcircuit $= c, d, e, f, b, c$

Example







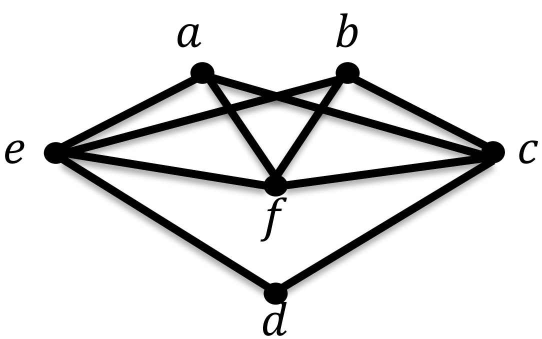
circuit $= a, b, e, a, c, d, e, f, b, c, f, a$

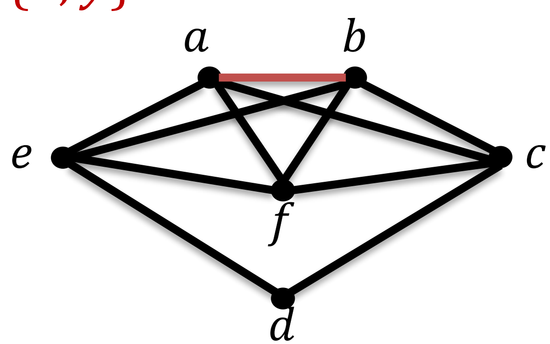
Euler Paths

THEOREM: Let $G = (V, E)$ be a connected multigraph of order ≥ 2. Then G has an Euler path (not Euler circuit) iff G has exactly 2 vertices of odd degree. 一起一终

ALGORITHM:

- **Input**: $G = (V, E)$, a connected multigraph, $x, y \in V$ have odd degrees
- **Output**: an Euler path
  - $H := G + \{x, y\}$
  - find an Euler circuit using Hierholzer's algorithm
  - remove the edge $\{x, y\}$ from the circuit





$a, c, d, e, f, b, a, e, b, c, f, a$

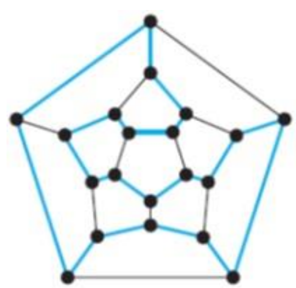
$a, c, d, e, f, b, a, e, b, c, f, a$

$a, e, b, c, f, a, c, d, e, f, b$

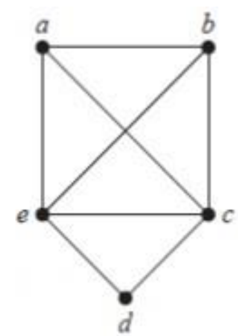
Hamilton Paths and Circuits

DEFINITION: Let $G = (V, E)$ be a graph.

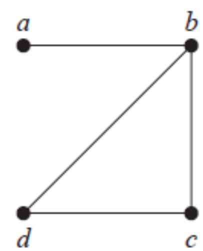
- **Hamilton Path**: A simple path that passes through every vertex exactly once.
- **Hamilton Circuit:** A simple circuit that passes through every vertex exactly once. (顶点 2次)



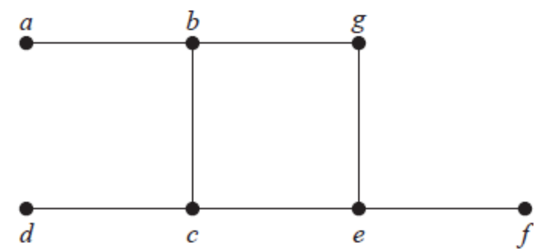
√ Hamilton path
√ Hamilton circuit



√ Hamilton path
√ Hamilton circuit



√ Hamilton path
×Hamilton circuit



× Hamilton path
× Hamilton circuit

Hamilton Circuits

Determine if there is a Hamilton circuit in a given graph G?

- This problem is NP-Complete. //that means very difficult

Necessary conditions on Hamilton circuit. 进后出不了

- If G has a vertex of degree 1, then G cannot have a Hamilton circuit.
- If G has a vertex of degree 2, then a Hamilton circuit of G traverses both edges. 一进一出

充分条件 $\Rightarrow$

Sufficient conditions on Hamilton circuit. (不满足也可能有Hamilton回路)

奥尔定理

- **Ore's Theorem:** Let $G = (V, E)$ be a simple graph of order $n \geq 3$. If $\deg(u) + \deg(v) \geq n$ for all $\{u, v\} \notin E$, then G has a Hamilton circuit.

狄拉克

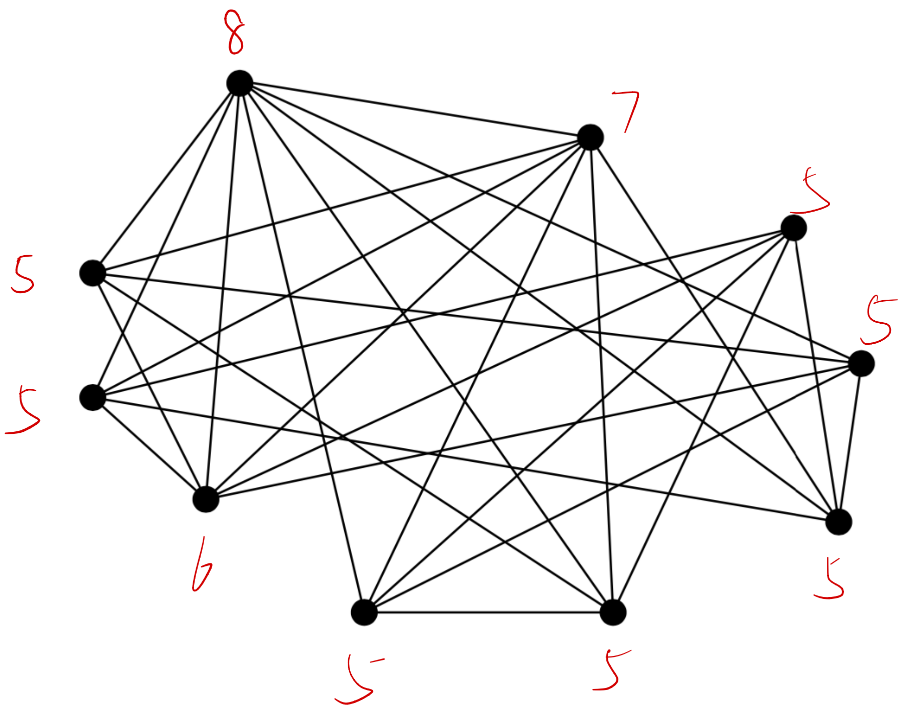
- **Dirac's Theorem:** Let $G = (V, E)$ be a simple graph of order $n \geq 3$. If $\deg(u) \geq n/2$ for every $u \in V$, then G has a Hamilton circuit.
  - This is a corollary of Ore's Theorem
    - $\forall u \in V,\ \deg(u) \geq n/2 \Rightarrow \forall u, v \in V, \deg(u) + \deg(v) \geq n$

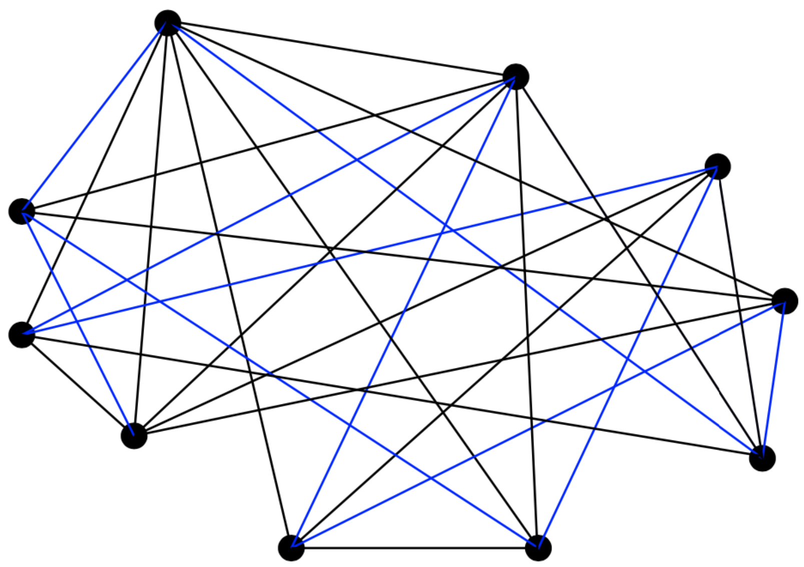
Hamilton Circuits

- Examples (sufficient condition)

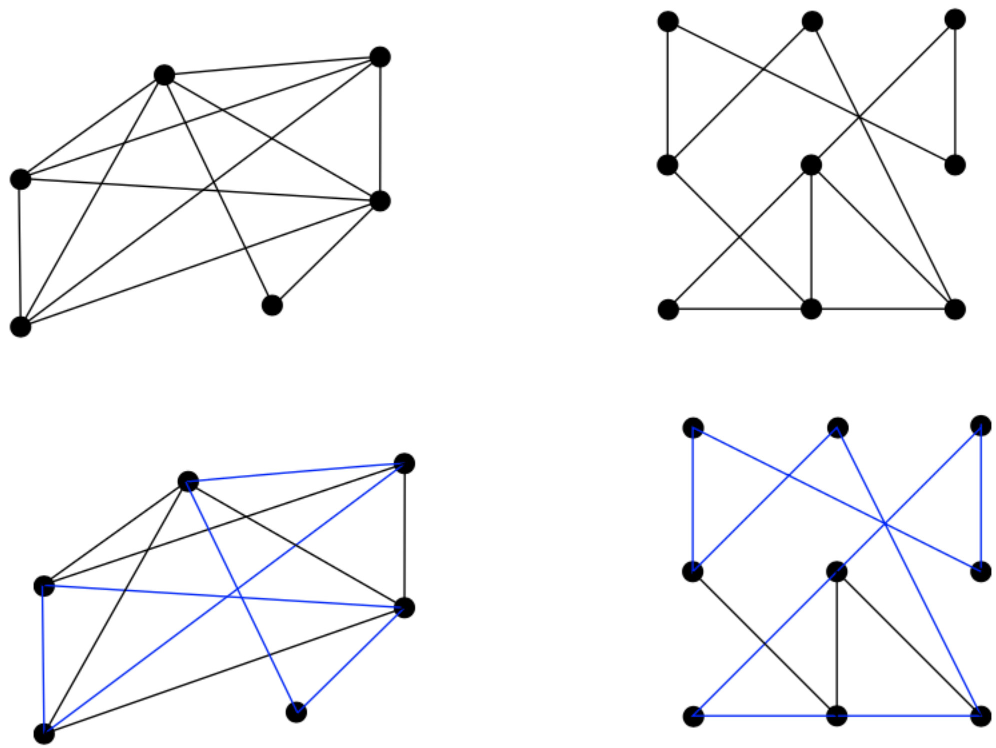
$n = 10$

$\min\{\deg(v)\} = 5$





Hamilton Circuits

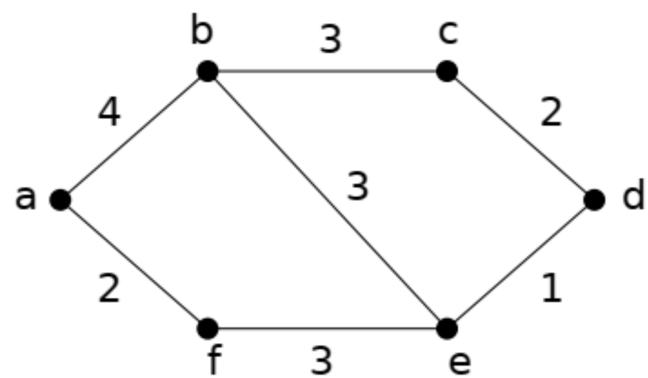


Remark: Dirac's and Ore's Theorems do not give a necessary condition for the existence of a Hamilton circuit!

Shortest Path Problem

Definition

A **weighted graph** is a graph $G = (V, E)$ such that each edge is assigned with a strictly positive number.
The **length** of a path in weighted graph is the sum of the weights of the edges of this path.

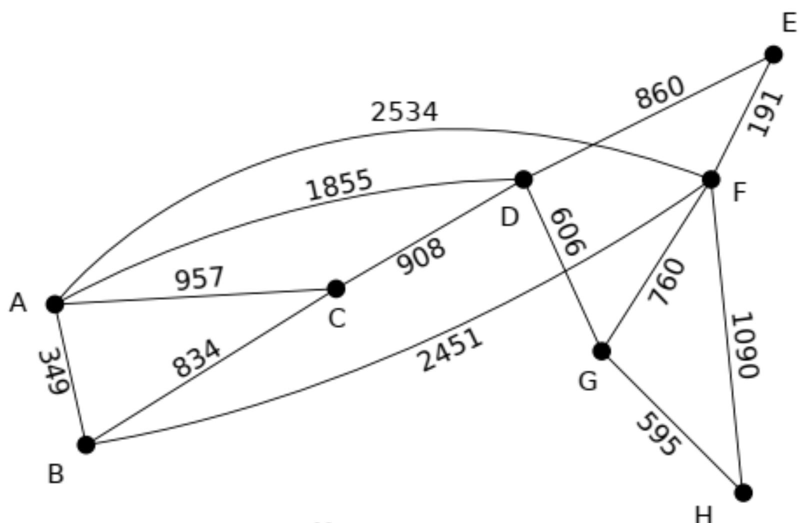


a, b, c is a path of length 7 and b, e, d, c is a path of length 6

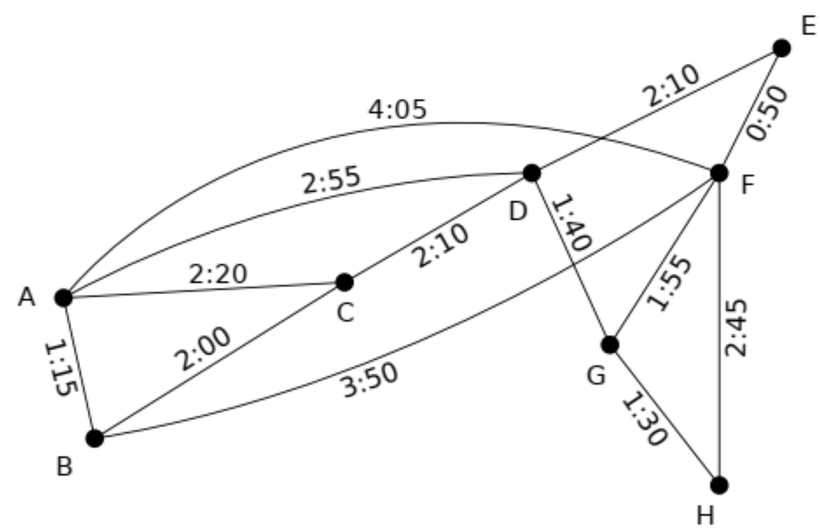
Remark: Observe that in a non-weighted graph the length of a path is the number of edges in the path!

Shortest Path Problem

Examples



distance



time

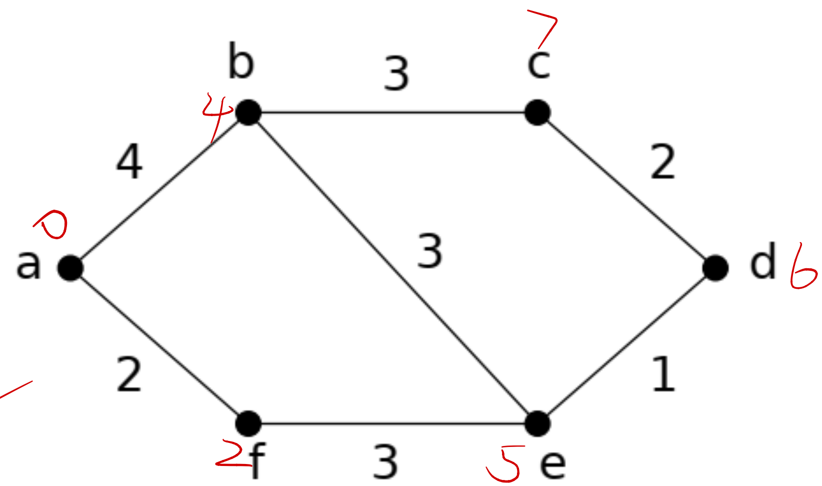
What is the shortest path in air distance between cities A and E?
What combination of flights has the smallest total flight time?

Shortest Path Problem

Question: Find the shortest path from a to d.

$\Rightarrow \begin{matrix}2\\4\end{matrix} \Rightarrow \begin{matrix}4\\5\end{matrix} \Rightarrow \begin{matrix}5\\7\end{matrix}$

$\Rightarrow \begin{matrix}6\\7\end{matrix} \Rightarrow 7 \Rightarrow$



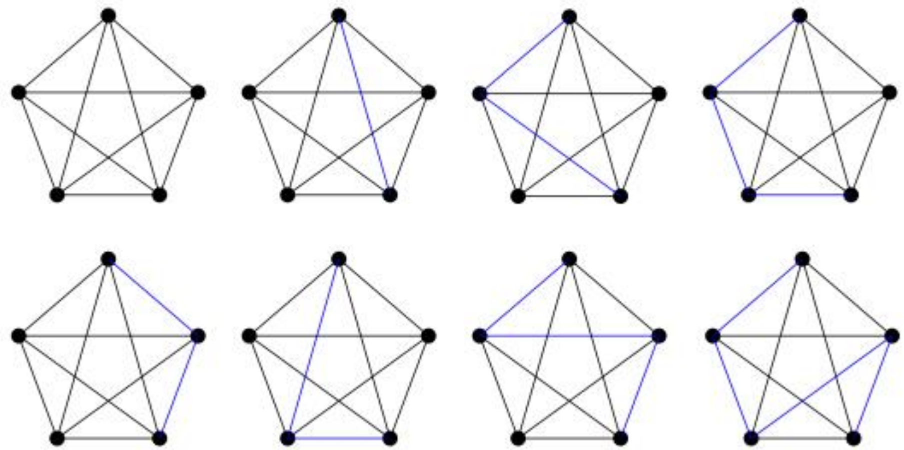
Method: Find the closest vertex to a, then the second closest, the third closest... until we reach d.

$\Rightarrow$ Dijkstra's algorithm

Shortest Path Problem

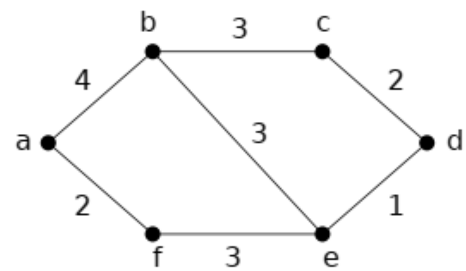
Remarks:

- Of course in the example above, we could have looked at all the paths between a and d and compute their length, but too complicated if the graph has a lot of edges.



- Advantage of Dijkstra's algorithm: we can compute the length of a shortest path from one vertex to all other vertices of the graph.

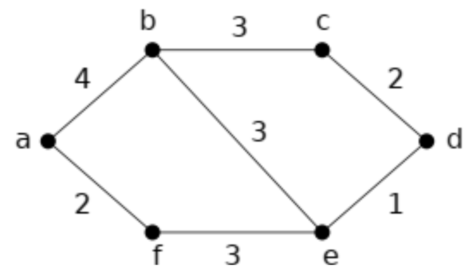
Djikstra's Algorithm



1. Find the closest vertex to $a \rightsquigarrow$ analyse all the edges starting from a:
   a, b of length 4
   a, f of length 2
   $\Rightarrow$ f is the closest vertex to a. The shortest path from a to f has length 2.

2. Find the second closest vertex to $a \rightsquigarrow$ shortest paths from a to a vertex in $\{a, f\}$ followed by an edge from a vertex in $\{a, f\}$ to a vertex not in this set:
   a, b of length 4
   a, f, e of length 5
   $\Rightarrow$ b is the second closest vertex to a. The shortest path from a to b has length 4.

Djikstra's Algorithm



3. Find the third closest vertex to $a \rightsquigarrow$ shortest path from a to a vertex in $\{a, f, b\}$ followed by an edge from a vertex in $\{a, f, b\}$ to a vertex not in this set:
   a, b, c of length 7
   a, b, e of length 7
   a, f, e of length 5
   $\Rightarrow$ e is the third closest vertex to a. The shortest path from a to e has length 5.

4. Find the fourth closest vertex to $a \rightsquigarrow$ shortest path from a to a vertex in $\{a, f, b, e\}$ followed by an edge from a vertex in $\{a, f, b, e\}$ to a vertex not in this set:
   a, b, c of length 7
   a, f, e, d of length 6
   $\Rightarrow$ d is the fourth closest vertex to a. The shortest path from a to d has length 6.

Djikstra's Algorithm

Goal: find the length of a shortest path from a to z with a series of iterations.

• A distinguished set of vertices is constructed by adding one vertex at each iteration.

• A labeling procedure is carried out at each iteration: a vertex w is labeled with the length of a shortest path from a to w that contains only vertices in the distinguished set.

• The vertex added to the distinguished set is one with minimal label among those vertices not already in the set.

Notations: S_k := distinguished set after k iterations, $L_k(v)$:= length of a shortest path from a to v containing only vertices in S_k ("label" of v).

Initialization: $L_0(a) = 0$,
$L_0(v) = \infty$ for every vertex $v \neq a$,
$S_0 = \emptyset$.

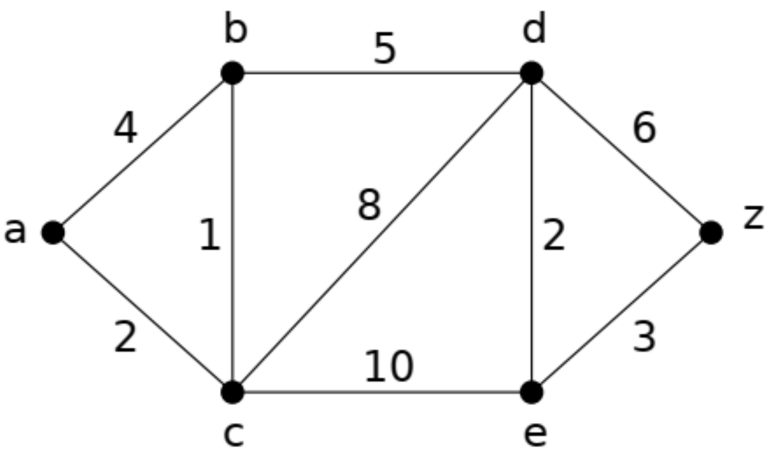
***k*th iteration:**

• S_k is formed from S_{k-1} by adding a vertex u not in S_{k-1} with smallest label,

• Update the labels of all vertices not in S_k so that $L_k(v)$ is the length of a shortest path from a to v containing only vertices in S_k, i.e.

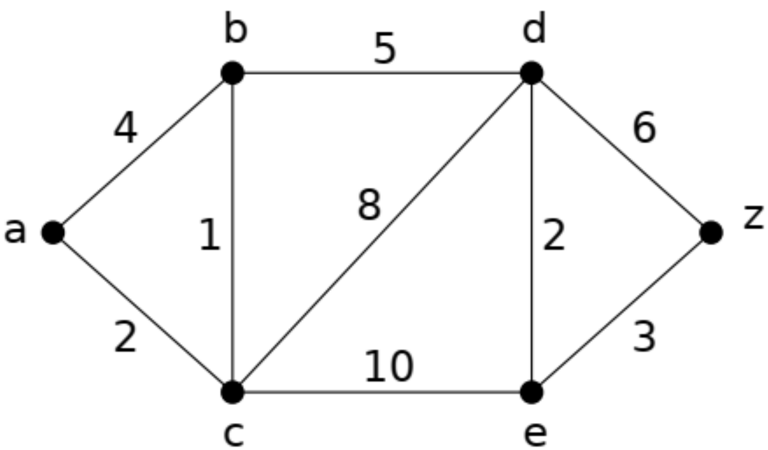
$L_k(v) = min\{L_{k-1}(v), L_{k-1}(u) + w(u, v)\}$ (with $w(u, v)$ length of the edge (u, v))

Djikstra's Algorithm



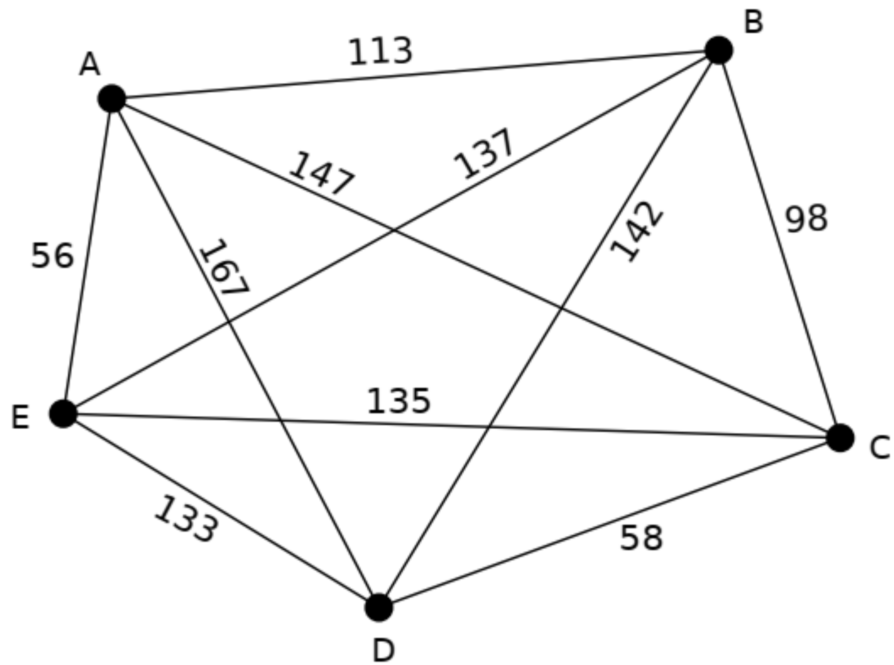
- **k=0 (initialization):** $S_0 = \emptyset$, $L_0(a) = 0,\ L_0(b) = L_0(c) = L_0(d) = L_0(e) = L_0(z) = \infty$

Djikstra's Algorithm



- **k=0 (initialization):** $S_0 = \emptyset$, $L_0(a) = 0,\ L_0(b) = L_0(c) = L_0(d) = L_0(e) = L_0(z) = \infty$
- **k=1:** $u := a \rightsquigarrow S_1 = \{a\}$,
  $L_0(a) + w(a,b) = 4 < L_0(b) \rightsquigarrow L_1(b) = 4$
  $L_0(a) + w(a,c) = 2 < L_0(c) \rightsquigarrow L_1(c) = 2$
- **k=2:** $u := c \rightsquigarrow S_1 = \{a, c\}$,
  $L_1(c) + w(c,b) = 3 < L_1(b) \rightsquigarrow L_2(b) = 3$
  $L_1(c) + w(c,d) = 10 < L_1(d) \rightsquigarrow L_2(d) = 10$
  $L_1(c) + w(c,e) = 12 < L_1(e) \rightsquigarrow L_2(e) = 12$
- **k=3:** $u := b \rightsquigarrow S_1 = \{a, c, b\}$,
  $L_2(b) + w(b,d) = 8 < L_2(d) \rightsquigarrow L_3(d) = 8$
- **k=4:** $u := d \rightsquigarrow S_1 = \{a, c, b, d\}$,
  $L_3(d) + w(d,e) = 10 < L_3(e) \rightsquigarrow L_4(e) = 10$
  $L_3(d) + w(d,z) = 14 < L_3(z) \rightsquigarrow L_4(z) = 14$
- **k=5:** $u := e \rightsquigarrow S_1 = \{a, c, b, d, e\}$,
  $L_4(e) + w(e,z) = 13 < L_4(z) \rightsquigarrow L_5(z) = 13$
- **k=6:** $u := z \rightsquigarrow S_1 = \{a, c, b, d, z\}$,
- **return:** $L(z) = 13$

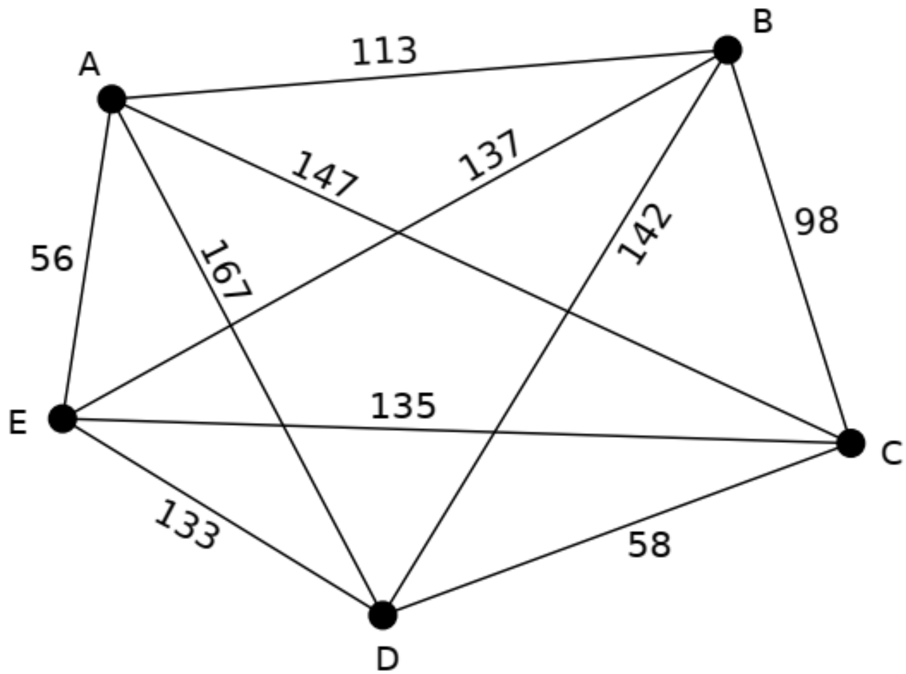
Traveling Salesperson Problem



Traveling salesperson problem: a traveling salesperson wants to visit each of the cities once and return to his starting point. In which order should he visit these cities to travel the minimum total distance?

$\Rightarrow$ **Hamiltonian circuit with minimum total weight in the complete graph.**

Traveling Salesperson Problem



Route	Tot. dist.
A, B, C, D, E, A	610
A, B, C, E, D, A	516
A, B, E, D, C, A	588
A, B, E, C, D, A	458
A, B, D, E, C, A	540
A, B, D, C, E, A	504
A, D, B, C, E, A	598
A, D, B, E, C, A	576
A, D, E, B, C, A	682
A, D, C, B, E, A	646
A, C, D, B, E, A	670
A, C, B, D, E, A	728

Traveling salesperson problem: a traveling salesperson wants to visit each of the cities once and return to his starting point. In which order should he visit these cities to travel the minimum total distance?
$\Rightarrow$ **Hamiltonian circuit with minimum total weight in the complete graph.**