Chapter 9
More About Graphs

Discrete Mathematics I on 7 May 2012

Tran Vinh Tan
Faculty of Computer Science and Engineering
University of Technology - VNUHCM

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring
Acknowledgement

Most of these slides were either created by **Dr. Huynh Tuong Nguyen** at University of Technology or else are modifications of his slides. Some slides about Euler and Hamilton circuits are created by Chung Ki-hong and Hur Joon-seok from KAIST.
Contents

1 Connectivity
   Paths and Circuits

2 Euler and Hamilton Paths
   Euler Paths and Circuits
   Hamilton Paths and Circuits

3 Shortest Path Problem
   Dijkstra’s Algorithm
   Bellman-Ford Algorithm
   Floyd-Warshall Algorithm
   Traveling Salesman Problem

4 Planar Graphs

5 Graph Coloring
Paths and Circuits

Simple path of length 4

Circuit of length 4

CuuDuongThanCong.com

https://fb.com/tailieudientucntt
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4:

Circuit of length 4:

https://fb.com/tailieudientucntt
Paths and Circuits

Simple path of length 4

Circuit of length 4

9.4

https://fb.com/tailieudientucntt
Paths and Circuits

Simple path of length 4
Paths and Circuits

Simple path of length 4
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4

Circuit of length 4
Paths and Circuits

Simple path of length 4

Euler and Hamilton Paths

Shortest Path Problem

Traveling Salesman Problem

Planar Graphs

Graph Coloring

https://fb.com/tailieudientucntt
Paths and Circuits

Simple path of length 4

Circuit of length 4
Path and Circuits

**Definition (in undirected graph)**

- **Path (đường đi)** of length \( n \) from \( u \) to \( v \): a sequence of \( n \) edges \( \{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\} \), where \( x_0 = u \) and \( x_n = v \).
- A path is a **circuit (chu trình)** if it begins and ends at the same vertex, \( u = v \).
- A path or circuit is **simple (đơn)** if it does not contain the same edge more than once.

---

**Simple path**

https://fb.com/tailieudientucntt
Path and Circuits

Definition (in undirected graph)

- **Path** ([đường đi](#)) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.

- A path is a **circuit** ([chu trình](#)) if it begins and ends at the same vertex, $u = v$.

- A path or circuit is **simple** ([đơn](#)) if it does not contain the same edge more than once.

![Diagram of simple path](https://fb.com/tailieudientucntt)
Path and Circuits

Definition (in undirected graph)

- **Path** (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.

- A path is a **circuit** (chu trình) if it begins and ends at the same vertex, $u = v$.

- A path or circuit is **simple** (đơn) if it does not contain the same edge more than once.

![Diagram](https://fb.com/tailieudientucntt)
Path and Circuits

Definition (in undirected graph)

- **Path** (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges \( \{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\} \), where $x_0 = u$ and $x_n = v$.
- A path is a **circuit** (chu trình) if it begins and ends at the same vertex, $u = v$.
- A path or circuit is **simple** (đơn) if it does not contain the same edge more than once.

![Simple path](https://fb.com/tailieudientucntt)
Path and Circuits

Definition (in undirected graph)

- **Path** (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.

- A path is a **circuit** (chu trình) if it begins and ends at the same vertex, $u = v$.

- A path or circuit is **simple** (dơn) if it does not contain the same edge more than once.

![Graph showing simple paths and circuits](https://fb.com/tailieudientucntt)
Path and Circuits

Definition (in undirected graph)

- Path (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.
- A path is a circuit (chu trình) if it begins and ends at the same vertex, $u = v$.
- A path or circuit is simple (đơn) if it does not contain the same edge more than once.
Path and Circuits

Definition (in undirected graph)

- **Path** (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.

- A path is a **circuit** (chu trình) if it begins and ends at the same vertex, $u = v$.

- A path or circuit is **simple** (đơn) if it does not contain the same edge more than once.

---

Simple path

https://fb.com/tailieudientucntt
Path and Circuits

Definition (in undirected graph)

- Path (đường đi) of length $n$ from $u$ to $v$: a sequence of $n$ edges $\{x_0, x_1\}, \{x_1, x_2\}, \ldots, \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$.
- A path is a circuit (chu trình) if it begins and ends at the same vertex, $u = v$.
- A path or circuit is simple (đơn) if it does not contain the same edge more than once.

![Diagram showing simple and not simple paths](https://fb.com/tailieudientucntt)
Path and Circuits

Definition (in directed graphs)

Path is a sequence of \((x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)\), where \(x_0 = u\) and \(x_n = v\).
Connectedness in Undirected Graphs

**Definition**

- An undirected graph is called **connected** (liên thông) if there is a path between every pair of distinct vertices of the graph.
- There is a simple path between every pair of distinct vertices of a connected undirected graph.
**Connectedness in Undirected Graphs**

**Definition**

- An undirected graph is called **connected** (*liên thông*) if there is a path between every pair of distinct vertices of the graph.
- There is a simple path between every pair of distinct vertices of a connected undirected graph.

Disconnected graph

https://fb.com/tailieudientucntt
Connectedness in Undirected Graphs

Definition

• An undirected graph is called **connected** (*liên thông*) if there is a path between every pair of distinct vertices of the graph.
• There is a simple path between every pair of distinct vertices of a connected undirected graph.

Connected components (*thành phần liên thông*)

![Connected components](https://fb.com/tailieudientucntt)
How Connected is a Graph?

Definition

- $b$ is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).

Contents

- Connectivity
- Paths and Circuits
- Euler and Hamilton Paths
- Shortest Path Problem
  - Dijkstra’s Algorithm
  - Bellman-Ford Algorithm
  - Floyd-Warshall Algorithm
- Traveling Salesman Problem
- Planar Graphs
- Graph Coloring

More About Graphs
Tran Vinh Tan

https://fb.com/tailieudientucntt
How Connected is a Graph?

Definition

- $b$ is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).
How Connected is a Graph?

Definition

• $b$ is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).

Contents

Connectivity

Paths and Circuits

Euler and Hamilton

Paths

Euler Paths and Circuits

Hamilton Paths and Circuits

Shortest Path Problem

Dijkstra's Algorithm

Bellman-Ford Algorithm

Floyd-Warshall Algorithm

Traveling Salesman Problem

Planar Graphs

Graph Coloring

CuuDuongThanCong.com

https://fb.com/tailieudientucntt
How Connected is a Graph?

Definition

- $b$ is a **cut vertex** (đỉnh cắt) or **articulation point** (điểm khớp).
  
What else?
How Connected is a Graph?

**Definition**

- **b** is a cut vertex (đỉnh cắt) or articulation point (diểm khớp).
- What else?
- \( \{a, b\} \) is a cut edge (cạnh cắt) or bridge (cầu).

---

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

---

cuu_duong_than_cong.com

9.8

ng.com  https://fb.com/tailieudientucntt
How Connected is a Graph?

Definition

- \( b \) is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).
  What else?
- \( \{a, b\} \) is a cut edge (cạnh cắt) or bridge (cầu).

Contents

- Connectivity
- Paths and Circuits
  - Euler and Hamilton Paths
  - Shortest Path Problem
    - Dijkstra’s Algorithm
    - Bellman-Ford Algorithm
    - Floyd-Warshall Algorithm
  - Traveling Salesman Problem
- Planar Graphs
- Graph Coloring

More About Graphs
Tran Vinh Tan

Definition

- \( b \) is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).
  What else?
- \( \{a, b\} \) is a cut edge (cạnh cắt) or bridge (cầu).
How Connected is a Graph?

Definition

- $b$ is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).
- What else?
- $\{a, b\}$ is a cut edge (cạnh cắt) or bridge (cầu).

Contents

Connectivity

Paths and Circuits

Euler and Hamilton Paths

Shortest Path Problem

Traveling Salesman Problem

Planar Graphs

Graph Coloring

More About Graphs

Tran Vinh Tan

How Connected is a Graph?

Definition

- $b$ is a cut vertex (đỉnh cắt) or articulation point (điểm khớp).
- What else?
- $\{a, b\}$ is a cut edge (cạnh cắt) or bridge (cầu).
How Connected is a Graph?

Definition

- \( b \) is a cut vertex (đỉnh cắt) or articulation point (diểm khớp).
- \( \{a, b\} \) is a cut edge (cạnh cắt) or bridge (cầu).

What else?
How Connected is a Graph?

Definition

- This graph has no cut vertices: nonseparable graph (đồ thị không thể phân tách).
- The vertex cut is \{c, f\}, so the minimum number of vertices in a vertex cut, vertex connectivity (liên thông đỉnh) \(\kappa(G) = 2\).
- The edge cut is \{\{b, c\}, \{a, f\}, \{f, g\}\}, the minimum number of edges in an edge cut, edge connectivity (liên thông cạnh) \(\lambda(G) = 3\).
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
- The vertex cut is \(\{c, f\}\), so the minimum number of vertices in a vertex cut, vertex connectivity (liên thông đỉnh) \(\kappa(G) = 2\).
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
- The vertex cut is \( \{c, f\} \), so the minimum number of vertices in a vertex cut, vertex connectivity (liên thông đỉnh) \( \kappa(G) = 2 \).
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
- The vertex cut is \( \{c, f\} \), so the minimum number of vertices in a vertex cut, vertex connectivity (liên thông đỉnh) \( \kappa(G) = 2 \).
How Connected is a Graph?

![Graph Diagram]

**Definition**

- This graph don't have cut vertices: **nonseparable graph** (độ thị không thể phân tách)
- The **vertex cut** is \( \{c, f\} \), so the minimum number of vertices in a vertex cut, **vertex connectivity** (liên thông đỉnh) \( \kappa(G) = 2 \).
How Connected is a Graph?

Definition

- This graph don’t have cut vertices: nonseparable graph (đồ thị không thể phân tách)
- The vertex cut is \{c, f\}, so the minimum number of vertices in a vertex cut, vertex connectivity (liên thông đỉnh) \(\kappa(G) = 2\).
- The edge cut is \{{b, c}, {a, f}, {f, g}\}, the minimum number of edges in an edge cut, edge connectivity (liên thông cạnh) \(\lambda(G) = 3\).
Applications of Vertex and Edge Connectivity

• Reliability of networks
  • Minimum number of routers that disconnect the network
  • Minimum number of fiber optic links that can be down to disconnect the network

• Highway network
  • Minimum number of intersections that can be closed
  • Minimum number of roads that can be closed
Connectedness in Directed Graphs

Definition

- An directed graph is strongly connected (*liên thông mạnh*) if there is a path between any two vertices in the graph (for both directions).
- An directed graph is weakly connected (*liên thông yếu*) if there is a path between any two vertices in the underlying undirected graph.

![Diagram of strongly and weakly connected graphs](https://fb.com/tailieudientucntt)
Applications

Example

Determine whether the graphs below are isomorphic.

\[ G \]

\[ H \]

Solution

\( H \) has a simple circuit of length three, not \( G \).
Applications

Example

Determine whether the graphs below are isomorphic.

Solution

Both graphs have the same vertices, edges, degrees, circuits. They may be isomorphic. To find a possible isomorphism, we can follow paths that go through all vertices so that the corresponding vertices in the two graphs have the same degrees.
The Famous Problem of Seven Bridges of Königsberg

- Is there a route that a person crosses all the seven bridges once?
Euler Solution

Euler gave the solution: It is not possible to cross all the bridges exactly once.

https://fb.com/tailieudientucntt
**Euler Solution**

Euler gave the solution: It is not possible to cross all the bridges exactly once.
Euler Solution

Euler gave the solution: It is not possible to cross all the bridges exactly once.
Euler Solution

Euler gave the solution: It is not possible to cross all the bridges exactly once.
**Euler Solution**

- Euler gave the solution: It is **not** possible to cross all the bridges exactly once.
What is Euler Path and Circuit?

- **Euler Path** (đường đi Euler) is a path in the graph that passes each edge only once.
- **Euler Circuit** (chu trình Euler) is a path in the graph that passes each edge only once and returns back to its original position.

From Definition, Euler Circuit is a subset of Euler Path.
What is Euler Path and Circuit?

- **Euler Path** (*đường đi Euler*) is a path in the graph that passes each edge only once.
What is Euler Path and Circuit?

- **Euler Path** (đường đi Euler) is a path in the graph that passes each edge only once. The problem of Seven Bridges of Königsberg can be also stated: Does Euler Path exist in the graph?

- **Euler Circuit** (chu trình Euler) is a path in the graph that passes each edge only once and return back to its original position. From Definition, Euler Circuit is a subset of Euler Path.
What is Euler Path and Circuit?

- **Euler Path** *(đường đi Euler)* is a path in the graph that passes each edge only once. The problem of Seven Bridges of Königsberg can be also stated: Does Euler Path exist in the graph?

- **Euler Circuit** *(chu trình Euler)* is a path in the graph that passes each edge only once and return back to its original position.
What is Euler Path and Circuit?

- **Euler Path** *(đường đi Euler)* is a path in the graph that passes each edge only once. The problem of Seven Bridges of Königsberg can be also stated: Does Euler Path exist in the graph?

- **Euler Circuit** *(chu trình Euler)* is a path in the graph that passes each edge only once and return back to its original position. From Definition, Euler Circuit is a subset of Euler Path.
Examples of Euler Path and Circuit
### Examples of Euler Path and Circuit

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>

Euler Circuit: `A → B → C → D → E` (Closed Path)

Euler Path: `A → B → C → D` (Open Path)

[Examples of Euler Path and Circuit](https://fb.com/tailieudientucntt)
Examples of Euler Path and Circuit

1 → 2

4             3

More About Graphs
Tran Vinh Tan

Contents
- Connectivity
- Paths and Circuits
- Euler and Hamilton Paths
- Euler Paths and Circuits
- Hamilton Paths and Circuits
- Shortest Path Problem
- Dijkstra’s Algorithm
- Bellman-Ford Algorithm
- Floyd-Warshall Algorithm
- Traveling Salesman Problem
- Planar Graphs
- Graph Coloring

https://fb.com/tailieudientucntt
Examples of Euler Path and Circuit

1  2
   3
   4

Euler Path

Euler Circuit
Examples of Euler Path and Circuit
Examples of Euler Path and Circuit

1 2

3 4
Examples of Euler Path and Circuit

Euler Circuit

1 2 3 4
Examples of Euler Path and Circuit

Euler Circuit

1 → 2 → 3 → 4 → 1

A → B → C → A

Euler Circuit

[Diagram of a square graph with labeled vertices and edges marked to form an Euler circuit]
Examples of Euler Path and Circuit

Euler Circuit

1 → 2 → A → B → C → D → 1
Examples of Euler Path and Circuit

Euler Circuit
Examples of Euler Path and Circuit

Euler Circuit
Examples of Euler Path and Circuit

Euler Circuit

1 2 3 4

A B C D
Examples of Euler Path and Circuit

Euler Circuit
Examples of Euler Path and Circuit

Euler Circuit

1 2 3 4

Euler Path

A B C D
Examples of Euler Path and Circuit

Euler Circuit

Euler Path

1 2

A

B

C

D

3 4
Conditions for Existence

In a connected multigraph,

- Euler Circuit existence: no odd-degree nodes exist in the graph.
- Euler Path existence: 2 or no odd-degree nodes exist in the graph.

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Conditions for Existence

In a connected multigraph,

- Euler Circuit existence: no odd-degree nodes exist in the graph.
Conditions for Existence

In a connected multigraph,

- Euler Circuit existence: **no odd-degree nodes exist** in the graph.
- Euler Path existence: **2 or no odd-degree nodes exist** in the graph.
Conditions for Existence

In a connected multigraph,

- Euler Circuit existence: **no odd-degree nodes exist** in the graph.
- Euler Path existence: **2 or no odd-degree nodes exist** in the graph.
Back to the Seven Bridges Problem

• Four vertices of odd degree
• No Euler circuit → cannot cross each bridge exactly once, and return to starting point
• No Euler path, either

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Back to the Seven Bridges Problem
Back to the Seven Bridges Problem
Back to the Seven Bridges Problem

- Four vertices of odd degree
Back to the Seven Bridges Problem

• Four vertices of odd degree
• No Euler circuit → cannot cross each bridge exactly once, and return to starting point
• No Euler path, either
Searching Euler Circuits and Paths – Fleury’s Algorithm

- Choose a random vertex (if circuit) or an odd degree vertex (if path)
- Pick an edge joined to another vertex so that it is not a cut edge unless there is no alternative
- Remove the chosen edge. The above procedure is repeated until all edges are covered.
Searching Euler Circuits and Paths – Fleury’s Algorithm

• Choose a random vertex (if circuit) or an odd degree vertex (if path)
Searching Euler Circuits and Paths – Fleury’s Algorithm

• Choose a random vertex (if circuit) or an odd degree vertex (if path)
• Pick an edge joined to another vertex so that it is not a cut edge unless there is no alternative
Searching Euler Circuits and Paths – Fleury’s Algorithm

- Choose a random vertex (if circuit) or an odd degree vertex (if path)
- Pick an edge joined to another vertex so that it is not a cut edge unless there is no alternative
- Remove the chosen edge. The above procedure is repeated until all edges are covered.
Searching Euler Circuits and Paths – Hierholzer’s Algorithm

• Choose a starting vertex and find a circuit
• As long as there exists a vertex \( v \) that belongs to the current tour but that has adjacent edges not part of the tour, start another circuit from \( v \)

More efficient algorithm, \( O(n) \).
Search the Euler Circuits and Paths – Hierholzer’s Algorithm

- Choose a starting vertex and find a circuit
Searching Euler Circuits and Paths – Hierholzer’s Algorithm

- Choose a starting vertex and find a circuit
- As long as there exists a vertex $v$ that belongs to the current tour but that has adjacent edges not part of the tour, start another circuit from $v$

More efficient algorithm, $O(n)$. 
Exercise

Example

Are these following graph Euler path (circuit)? If yes, find one.
Exercise

Example
Are these following graph Euler path (circuit)? If yes, find one.
Traveling Salesman Problem
Traveling Salesman Problem

Is there the possible tour that visits each city exactly once?

https://fb.com/tailieudientucntt
Traveling Salesman Problem
Traveling Salesman Problem

Is there the possible tour that visits each city exactly once?
What Is A Hamilton Circuit?

Definition
The circuit that visit each vertex in a graph once.
What Is A Hamilton Circuit?

Definition

The circuit that visit each vertex in a graph once
Definition

The circuit that visit each vertex in a graph **once**
**What Is A Hamilton Circuit?**

**Definition**

The circuit that visit each vertex in a graph *once*
What Is A Hamilton Circuit?

Definition

The circuit that visit each vertex in a graph once
What Is A Hamilton Circuit?

**Definition**

The circuit that visit each vertex in a graph once.
What Is A Hamilton Circuit?

**Definition**

The circuit that visit each vertex in a graph once
Rules of Hamilton Circuits

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]
Rules of Hamilton Circuits

\[ \deg(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \deg(v) = 2 \), both edge must be used.
Rules of Hamilton Circuits

\[ \deg(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \deg(v) = 2 \), both edge must be used.

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]
Rules of Hamilton Circuits

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \text{deg}(v) = 2 \), both edge must be used.

\( v \)
Rules of Hamilton Circuits

deg(v) = 2 for ∀v in Hamilton circuit!

Rule 1  if deg(v) = 2, both edge must be used.

Rule 2  No subcircuit (chu trình con) can be formed.
Rules of Hamilton Circuits

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \text{deg}(v) = 2 \), both edge must be used.

**Rule 2** No subcircuit \((chu \ trinh \ con)\) can be formed.

**Rule 3** Once two edges at a vertex \( v \) is determined, all other edges incident at \( v \) must be removed.
Rules of Hamilton Circuits

$$\deg(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!}$$

**Rule 1** if $$\deg(v) = 2$$, both edge must be used.

**Rule 2** No subcircuit (chu trình con) can be formed.

**Rule 3** Once two edges at a vertex $$v$$ is determined, all other edges incident at $$v$$ must be removed.
Rules of Hamilton Circuits

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \text{deg}(v) = 2 \), both edge must be used.

**Rule 2** No subcircuit (\( chu \ trinh \ con \)) can be formed.

**Rule 3** Once two edges at a vertex \( v \) is determined, all other edges incident at \( v \) must be removed.
Rules of Hamilton Circuits

\[ \text{deg}(v) = 2 \text{ for } \forall v \text{ in Hamilton circuit!} \]

**Rule 1** if \( \text{deg}(v) = 2 \), both edge must be used.

**Rule 2** No subcircuit (\( chu \ tinh \ con \)) can be formed.

**Rule 3** Once two edges at a vertex \( v \) is determined, all other edges incident at \( v \) must be removed.
Finding Hamilton Circuits

Vertices : cities
Edges : possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed

We get Hamilton circuit!

https://fb.com/tailieudientucntt
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1: \( \text{deg}(v) = 2 \)
Rule 2: Once two edges are determined, other edges must be removed.
We get Hamilton circuit!
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
$$\deg(v) = 2$$
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed.
We get Hamilton circuit!
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\( \text{deg}(v) = 2 \)
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]
### Finding Hamilton Circuits

Vertices: cities  
Edges: possible routes

**Rule 1**  
\[ \text{deg}(v) = 2 \]

**Rule 3**  
Once two edges are determined, other edges must be removed.
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \deg(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \deg(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed

Vertices: cities
Edges: possible routes
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\( \deg(v) = 2 \)

Rule 3
Once two edges are determined, other edges must be removed

Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra's Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\( \deg(v) = 2 \)

Rule 3
Once two edges are determined, other edges must be removed

Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra's Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\( \text{deg}(v) = 2 \)

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed

We get Hamilton circuit!
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
$$\text{deg}(v) = 2$$

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\[ \text{deg}(v) = 2 \]

Rule 3
Once two edges are determined, other edges must be removed
Finding Hamilton Circuits

Vertices: cities
Edges: possible routes

Rule 1
\( \text{deg}(v) = 2 \)

Rule 3
Once two edges are determined, other edges must be removed

We get Hamilton circuit!
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph.
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not.
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not.

Simple check by rules of Hamilton circuit
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not.

Simple check by rules of Hamilton circuit
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not. Simple check by rules of Hamilton circuit.
Existence of Hamilton Circuit

Hamilton circuit does not exist for all graph. But, there is no specific way to find whether Hamilton circuit exists or not.

Simple check by rules of Hamilton circuit

Violates Rule 2! (No subcircuit)
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify *nonexistence* of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify **nonexistence** of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify **nonexistence** of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify **nonexistence** of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify **nonexistence** of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify **nonexistence** of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.

We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.
We can verify nonexistence of the graph during find Hamilton circuit.

Contradict with Rule 1!
We can verify nonexistence of the graph during find Hamilton circuit.

Contradict with Rule 1!

Hamilton circuit doesn’t exist!
## Application – Gray Code

**Definition**

The **binary sequence** that express consecutive numbers by differing just one position of sequence.

<table>
<thead>
<tr>
<th>Decimal number</th>
<th>Binary number</th>
<th>Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>010</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>011</td>
</tr>
</tbody>
</table>

...  

Used at digital communication for reduce the effect of noise; it prevents serious changes of information by noise.
Application – Gray Code

**Definition**

The *binary sequence* that express consecutive numbers by differing just one position of sequence.

<table>
<thead>
<tr>
<th>Decimal number</th>
<th>Binary number</th>
<th>Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>010</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>011</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### Application – Gray Code

**Definition**

The **binary sequence** that express consecutive numbers by differing **just one** position of sequence.

<table>
<thead>
<tr>
<th>Decimal number</th>
<th>Binary number</th>
<th>Gray code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>010</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>011</td>
</tr>
</tbody>
</table>

Used at **digital communication** for reduce the effect of noise; it prevents serious changes of information by noise.
Gray Code

An n-digit gray code can be generated by finding Hamilton circuits of n-dimensional hypercube!
Gray Code

$n$-digit gray code can be generated by finding Hamilton circuits of $n$-dimensional hypercube! Consider the case $n = 3$. 
Gray Code

\(n\)-digit gray code can be generated by finding Hamilton circuits of \(n\)-dimensional hypercube! Consider the case \(n = 3\).
Gray Code

\(n\)-digit gray code can be generated by finding Hamilton circuits of \(n\)-dimensional hypercube! Consider the case \(n = 3\).

![Graph]

Coordinate of each vertex is 3-digit binary sequences.
Gray Code

\(n\)-digit gray code can be generated by finding Hamilton circuits of \(n\)-dimensional hypercube! Consider the case \(n = 3\).

![Diagram of 3-dimensional hypercube with Hamiltonian paths]

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just one place.
Gray Code

An $n$-digit gray code can be generated by finding Hamilton circuits of an $n$-dimensional hypercube! Consider the case $n = 3$.

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just one place.

https://fb.com/tailieudientucntt
Gray Code

An $n$-digit gray code can be generated by finding Hamilton circuits of an $n$-dimensional hypercube! Consider the case $n = 3$.

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just on place.
Gray Code

An $n$-digit gray code can be generated by finding Hamilton circuits of an $n$-dimensional hypercube! Consider the case $n = 3$.

![Graph of a 3-dimensional hypercube](image)

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just one place. Hamilton circuits of a cubic graph makes the order of binary sequences!
**Gray Code**

An $n$-digit gray code can be generated by finding Hamilton circuits of $n$-dimensional hypercube! Consider the case $n = 3$.

![Diagram of a 3-dimensional hypercube with Hamiltonian circuits]

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just on place. Hamilton circuits of a cubic graph makes the order of binary sequences!
Gray Code

An $n$-digit gray code can be generated by finding Hamilton circuits of $n$-dimensional hypercube! Consider the case $n = 3$.

Coordinate of each vertex is 3-digit binary sequences. Coordinates of adjacent vertices differ in just one place. Hamilton circuits of a cubic graph makes the order of binary sequences!
Weighted Graphs

HO CHI MINH

Can Tho

Con Dao

Nha Trang

Da Nang

Hai Phong

HA NOI

Dien Bien

296

88

264

605

436

886

230

322

264

1146

126

88

610

610

1146

88
Problem

The problem is also sometimes called the single-pair shortest path problem, to distinguish it from the following generalizations:

- The **single-source shortest path problem**, in which we have to find shortest paths from a source vertex $v$ to all other vertices in the graph.
- The **single-destination shortest path problem**, in which we have to find shortest paths from all vertices in the graph to a single destination vertex $v$. This can be reduced to the single-source shortest path problem by reversing the edges in the graph.
- The **all-pairs shortest path problem**, in which we have to find shortest paths between every pair of vertices $v, v'$ in the graph.

These generalizations have significantly more efficient algorithms than the simplistic approach of running a single-pair shortest path algorithm on all relevant pairs of vertices.
Dijkstra’s Algorithm

procedure Dijkstra(G, a)
    // Initialization Step
    forall vertices v
        Label[v] := ∞
        Prev[v] := -1
    endfor
    Label(a) := 0 // a is the source node
    S := ∅
    // Iteration Step
    while z /∈ S
        u := a vertex not in S with minimal Label
        S := S ∪ {u}
        forall vertices v not in S
            if (Label[u] + Wt(u, v)) < Label(v)
                then begin
                    Label[v] := Label[u] + Wt(u, v)
                    Pred[v] := u
                end
    endwhile
Example

Example of a graph:

```
S | a  b  c  d  e  z
---|-------
∅ | 0 ∞ ∞ ∞ ∞ ∞
```

Example of a shortest path table:

```
S | a  b  c  d  e  z
---|-------
∅ | 0 ∞ ∞ ∞ ∞ ∞
```
Example

![Graph Diagram]

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

https://fb.com/tailieudientucntt
Example

\[
\begin{array}{c|ccccccc}
S & a & b & c & d & e & z \\
\hline
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
 a & 0 & \\
\end{array}
\]
Example

\[
\begin{array}{ccccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & & & & & \\
\end{array}
\]
### Example

The graph below shows a network of nodes and edges. The table below represents the distances between these nodes.

**Graph:**

![Graph Diagram]

**Table:**

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

Graph with vertices labeled a, b, c, d, e, z and edges with weights 1, 2, 3, 4, 5, 6, 8, 10, 2, 3, 12, 14.

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>
Example

\[ \begin{array}{cccccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & & & & \\
\end{array} \]

https://fb.com/tailieudientucntt
Example

\[
\begin{array}{c|ccccccc}
S & a & b & c & d & e & z \\
\hline
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
\end{array}
\]
Example

\begin{table}[h]
\centering
\begin{tabular}{c|cccccc}
S & a & b & c & d & e & z \\
\hline
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
\emptyset & 0 & 4 & 2 & \infty & \infty & \infty \\
\end{tabular}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example_graph.png}
\end{figure}
Example

\[ S \quad a \quad b \quad c \quad d \quad e \quad z \]

\[ \emptyset \quad 0 \quad \infty \quad \infty \quad \infty \quad \infty \quad \infty \]

\[ a \quad 0 \quad 4 \quad 2 \quad \infty \quad \infty \quad \infty \quad \infty \]

\[ c \quad 0 \quad 2 \]

https://fb.com/tailieudientucntt
Example

The graph shown above represents a network with nodes and edges. The table below illustrates the shortest path problem using the Dijkstra's algorithm. The table shows the distances between nodes, with infinity (∞) representing a non-existent path.

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The distances in the table are as follows:
- a to b: 4
- a to c: 2
- a to d: ∞
- a to e: ∞
- a to z: ∞
Example


\[
\begin{array}{lcccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
 a & 0 & 4 & 2 & \infty & \infty & \infty \\
 c & 0 & 3 & 2 & & & \\
\end{array}
\]
Example

### Graph Connectivity

#### Paths and Circuits

- Euler and Hamilton Paths
- Euler Paths and Circuits
- Hamilton Paths and Circuits

#### Shortest Path Problem

- Dijkstra's Algorithm
- Bellman-Ford Algorithm
- Floyd-Warshall Algorithm
- Traveling Salesman Problem

#### Planar Graphs

#### Graph Coloring

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: CuuDuongThanCong.com

Website: https://fb.com/tailieudientucntt
Example

```
S   a   b   c   d   e   z
∅   0   ∞   ∞   ∞   ∞   ∞
a   0   4   2   ∞   ∞   ∞
c   0   3   2   10  ∞   ∞
```

```
Example

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra's Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

S   a   b   c   d   e   z
∅   0   ∞   ∞   ∞   ∞   ∞
a   0   4   2   ∞   ∞   ∞
c   0   3   2   10  ∞   ∞
```
Example

<table>
<thead>
<tr>
<th>$S$</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$0$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$a$</td>
<td>$0$</td>
<td>$4$</td>
<td>$2$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$c$</td>
<td>$0$</td>
<td>$3$</td>
<td>$2$</td>
<td>$10$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>
### Example

```
\begin{array}{ccccccc}
\text{S} & \text{a} & \text{b} & \text{c} & \text{d} & \text{e} & \text{z} \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
\end{array}
```

![Graph Diagram](attachment:graph.png)

- **Connectivity**
- **Paths and Circuits**
- **Euler and Hamilton**
- **Paths**
- **Euler Paths and Circuits**
- **Hamilton Paths and Circuits**
- **Shortest Path Problem**
  - **Dijkstra's Algorithm**
  - **Bellman-Ford Algorithm**
  - **Floyd-Warshall Algorithm**
- **Traveling Salesman Problem**
- **Planar Graphs**
- **Graph Coloring**
### Example

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
</tbody>
</table>

The graph shows a network of cities with distances between them. The table represents the shortest path from city a to other cities, with infinite distances indicating paths that are not possible due to distance constraints.
Example

\begin{table}
\begin{tabular}{c|ccccccc}
S & a     & b     & c     & d     & e     & z     \\
\hline
\emptyset & 0     & \infty & \infty & \infty & \infty & \infty \\
\{a\}     & 0     & 4     & 2     & \infty & \infty & \infty \\
\{c\}     & 0     & 3     & 2     & 10    & 12    & \infty \\
\end{tabular}
\end{table}
Example

```
\[
\begin{array}{c|ccccccc}
S  & a  & b  & c  & d  & e  & z \\
\hline
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
 a     & 0 & 4 & 2 & \infty & \infty & \infty \\
 c     & 0 & 3 & 2 & 10  & 12  & \infty \\
 b     & 0 & 3 & 2 &     &     &     \\
\end{array}
\]
```
Example

<table>
<thead>
<tr>
<th>$S$</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>0</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$a$</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$c$</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$b$</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example

\begin{tabular}{c|cccccc}
  \( S \) & \( a \) & \( b \) & \( c \) & \( d \) & \( e \) & \( z \) \\
  \hline
  \( \emptyset \) & 0 & \( \infty \) & \( \infty \) & \( \infty \) & \( \infty \) & \( \infty \) \\
  \( a \) & 0 & 4 & 2 & \( \infty \) & \( \infty \) & \( \infty \) \\
  \( c \) & 0 & 3 & 2 & 10 & 12 & \( \infty \) \\
  \( b \) & 0 & 3 & 2 & 8 & & \\
\end{tabular}
Example

\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$S$ & $a$ & $b$ & $c$ & $d$ & $e$ & $z$ \\
\hline
$\emptyset$ & 0 & $\infty$ & $\infty$ & $\infty$ & $\infty$ & $\infty$ \\
$a$ & 0 & 4 & 2 & $\infty$ & $\infty$ & $\infty$ \\
$c$ & 0 & 3 & 2 & 10 & 12 & $\infty$ \\
$b$ & 0 & 3 & 2 & 8 & 12 & $\infty$ \\
\hline
\end{tabular}
### Example

Consider the graph shown below.

```
     a
    /  \
   /    \
  4     5
   \    / \
    \  b   d
       |   |
     1   8 
       |   |
  2       6
     c  e  z
```

The edges and their weights are as follows:
- Edge (a, b): weight 4
- Edge (b, c): weight 1
- Edge (c, d): weight 8
- Edge (d, e): weight 2
- Edge (e, z): weight 3

#### Table

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
</tbody>
</table>
Example

\[
\begin{array}{cccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & & \\
\end{array}
\]
Example

\[ \begin{array}{cccccc}
S & a & b & c & d & e \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & \infty & \infty \\
\end{array} \]
Example

\[ \begin{array}{cccccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & 10 & \\
\end{array} \]
Example

\[
\begin{array}{cccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & 10 & \\
\end{array}
\]
Example

\[ \begin{array}{ccccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
av & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & 10 & 14 \\
\end{array} \]
Example

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra's Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Example

![Graph Diagram]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
</tbody>
</table>

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Example

Table:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
Example

- **Graph Representation:**

```
  a -- 1 -- 8 -- d
  |      |      |
  4     5     6
  v      v      v
  2 -- 10 -- 12
      d

- **Table Representation:**

<table>
<thead>
<tr>
<th>S</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
```

- **Notes:**

- Euler and Hamilton Paths
- Shortest Path Problem
- Dijkstra’s Algorithm
- Bellman-Ford Algorithm
- Floyd-Warshall Algorithm
- Traveling Salesman Problem

**References:**

- cuu duong than cong . com
- https://fb.com/tailieudientucntt
Example

```
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>a</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>c</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>10</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>∞</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>10</td>
<td>13</td>
</tr>
</tbody>
</table>
```
Example

\[
\begin{array}{cccccc}
S & a & b & c & d & e & z \\
\emptyset & 0 & \infty & \infty & \infty & \infty & \infty \\
a & 0 & 4 & 2 & \infty & \infty & \infty \\
c & 0 & 3 & 2 & 10 & 12 & \infty \\
b & 0 & 3 & 2 & 8 & 12 & \infty \\
d & 0 & 3 & 2 & 8 & 10 & 14 \\
e & 0 & 3 & 2 & 8 & 10 & 13 \\
\end{array}
\]
Dijkstra’s Algorithm

Property

Applicable for any $G$, any length $\ell(v_i) \geq 0$, $\forall i$; one-to-all; complexity $O(|V|^2)$.
Exercise

Example

Find the shortest path from $e$ to other vertices using Dijkstra’s algorithm.
Can Dijkstra’s Algorithm be used on...

- ...digraph?
  - Yes!
- ...negative weighted graph?
  - No! Why?

Dijkstra’s Algorithm Flaw
Bellman-Ford Algorithm

procedure BellmanFord(G, a)
// Initialization Step
forall vertices v
  Label[v] := ∞
  Prev[v] := -1
Label(a) := 0 // a is the source node

// Iteration Step
for i from 1 to size(vertices)-1
  forall vertices v
    if (Label[u] + Wt(u,v)) < Label[v]
      then
        Label[v] := Label[u] + Wt(u,v)
        Prev[v] := u

// Check circuit of negative weight
forall vertices v
  if (Label[u] + Wt(u,v)) < Label(v)
    error "Contains circuit of negative weight"

Property
any $G$, any length; one-to-all; detect whether there exists a circle of negative length; complexity $O(|V| 	imes |E|)$. 
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>∞</td>
<td>3</td>
<td>-5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
</tbody>
</table>

Stop since Step 4 = Step 3.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

Example

Example

Step
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
</tbody>
</table>

Example

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>∞</td>
<td>-5</td>
<td>3</td>
</tr>
</tbody>
</table>

Stop since Step 4 = Step 3.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>∞</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
</tbody>
</table>

Stop since Step 4 = Step 3.

Example

![Graph with edge weights](https://example.com/graph.png)
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>∞</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
</tbody>
</table>

Stop since Step 4 = Step 3.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>∞</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-2</td>
<td>3</td>
<td>-4</td>
<td>-5</td>
<td>3</td>
</tr>
</tbody>
</table>

Stop since Step 4 = Step 3.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 $\neq$ Step 5.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 \( \neq \) Step 5.
### Example

<table>
<thead>
<tr>
<th>Step</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 \( \neq \) Step 5.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 ≠ Step 5.
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 ≠ Step 5.
### Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 ≠ Step 5.

Example graph with weights:

```
   a --4--- d
   |    /   |
   --3--- |
   b /    |
   | -2--- f
   /    |
   c    e
```

---

Example:

```
<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>
```

Example graph with weights:

```
   a --4--- d
   |    /   |
   --3--- |
   b /    |
   | -2--- f
   /    |
   c    e
```
### Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 \( \neq \) Step 5.

![Graph Diagram](https://example.com/graph.png)
Example

<table>
<thead>
<tr>
<th>Step</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step 6 ≠ Step 5.

Example

```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>∞</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-6</td>
<td>-3</td>
</tr>
</tbody>
</table>
```
## Example

### Table

<table>
<thead>
<tr>
<th>Step</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>$\infty$</td>
<td>-1</td>
<td>$\infty$</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-1</td>
<td>-4</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>6</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-6</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>

There exists a circle of negative length since Step $6 \neq$ Step $5$.

### Diagram

![Graph Diagram](https://fb.com/tailieudientucntt)
Exercise
Floyd-Warshall Algorithm [1962]

procedure FloydWarshall ()
    for k := 1 to n
        for i := 1 to n
            for j := 1 to n
                path[i,j] = min (path[i,j],
                                path[i,k]+path[k,j]);

Property
any $G$, any length; all-to-all; this is a software algorithm; complexity $O(|V|^3)$.
Example

\[ L^{(0)} = \begin{pmatrix}
  0 & 1 & \infty & 4 \\
  2 & 0 & -2 & \infty \\
  3 & \infty & 0 & \infty \\
  \infty & -5 & -1 & 0
\end{pmatrix} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & \infty \\ 3 & \infty & 0 & \infty \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(1)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & \infty & 0 & \infty \\ 3 & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty \end{pmatrix} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & \infty \\ 3 & \infty & 0 & \infty \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(1)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ \infty & -5 & -1 & 0 \end{pmatrix} \]
Example

\[ \begin{align*}
L^{(0)} &= \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & \infty \\
3 & \infty & 0 & \infty \\
\infty & -5 & -1 & 0 \\
\end{pmatrix} \\
L^{(1)} &= \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0 \\
\end{pmatrix} \\
L^{(2)} &= \begin{pmatrix}
2 & 0 & -2 & 6 \\
4 & 0 & -5 \\
\end{pmatrix}
\end{align*} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(1)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(2)} = \begin{pmatrix} 0 & 1 & -1 & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ -3 & -5 & -7 & 0 \end{pmatrix} \]
Example

```
\begin{align*}
\mathbf{L}^{(0)} &= \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0
\end{pmatrix}, \\
\mathbf{L}^{(1)} &= \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0
\end{pmatrix}, \\
\mathbf{L}^{(2)} &= \begin{pmatrix}
0 & 1 & -1 & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
-3 & -5 & -7 & 0
\end{pmatrix}, \\
\mathbf{L}^{(3)} &= \begin{pmatrix}
-1 & 0 & 2 & 7 \\
3 & 4 & 0 & 7 \\
-7 & -3 & -5 & -7 & 0
\end{pmatrix}
\end{align*}
```
Example

\[
L^{(0)} = \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0
\end{pmatrix}
\]

\[
L^{(1)} = \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0
\end{pmatrix}
\]

\[
L^{(2)} = \begin{pmatrix}
0 & 1 & -1 & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
-3 & -5 & -7 & 0
\end{pmatrix}
\]

\[
L^{(3)} = \begin{pmatrix}
0 & 1 & -1 & 4 \\
1 & 0 & -2 & 5 \\
3 & 4 & 0 & 7 \\
-4 & -5 & -7 & 0
\end{pmatrix}
\]
Example

\[
L^{(0)} = \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0 \\
\end{pmatrix}
\]

\[
L^{(1)} = \begin{pmatrix}
0 & 1 & \infty & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
\infty & -5 & -1 & 0 \\
\end{pmatrix}
\]

\[
L^{(2)} = \begin{pmatrix}
0 & 1 & -1 & 4 \\
2 & 0 & -2 & 6 \\
3 & 4 & 0 & 7 \\
3 & 4 & 0 & 7 \\
\end{pmatrix}
\]

\[
L^{(3)} = \begin{pmatrix}
0 & 1 & -1 & 4 \\
1 & 0 & -2 & 5 \\
3 & 4 & 0 & 7 \\
-4 & -5 & -7 & 0 \\
\end{pmatrix}
\]

\[
L^{(4)} = \begin{pmatrix}
0 & 1 & -1 & 4 \\
1 & 0 & -2 & 5 \\
3 & 4 & 0 & 7 \\
-4 & -5 & -7 & 0 \\
\end{pmatrix}
\]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(1)} = \begin{pmatrix} 0 & 1 & \infty & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ \infty & -5 & -1 & 0 \end{pmatrix} \]

\[ L^{(2)} = \begin{pmatrix} 0 & 1 & -1 & 4 \\ 2 & 0 & -2 & 6 \\ 3 & 4 & 0 & 7 \\ -3 & -5 & -7 & 0 \end{pmatrix} \]

\[ L^{(3)} = \begin{pmatrix} 0 & 1 & -1 & 4 \\ 1 & 0 & -2 & 5 \\ 3 & 4 & 0 & 7 \\ -4 & -5 & -7 & 0 \end{pmatrix} \]

\[ L^{(4)} = \begin{pmatrix} 0 & 1 & -1 & 4 \\ 1 & 0 & -2 & 5 \\ 3 & 2 & 0 & 7 \\ -4 & -5 & -7 & 0 \end{pmatrix} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 2 & \infty \\ 3 & 0 & -4 \\ 1 & \infty & 0 \end{pmatrix} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 2 & \infty \\ 3 & 0 & -4 \\ 1 & \infty & 0 \end{pmatrix} \quad L^{(1)} = \begin{pmatrix} 0 & 2 & \infty \\ 3 & 0 & -4 \\ 1 & 3 & 0 \end{pmatrix} \]
Example

\[ L^{(0)} = \begin{pmatrix} 0 & 2 & \infty \\ 3 & 0 & -4 \\ 1 & \infty & 0 \end{pmatrix} \quad L^{(1)} = \begin{pmatrix} 0 & 2 & \infty \\ 3 & 0 & -4 \\ 1 & 3 & 0 \end{pmatrix} \]

\[ L^{(2)} = \begin{pmatrix} 0 & 2 & -2 \\ 3 & 0 & -4 \\ 1 & 3 & -1 \end{pmatrix} \]
Example

\[
L^{(0)} = \begin{pmatrix}
0 & 2 & \infty \\
3 & 0 & -4 \\
1 & \infty & 0
\end{pmatrix} \quad L^{(1)} = \begin{pmatrix}
0 & 2 & \infty \\
3 & 0 & -4 \\
1 & 3 & 0
\end{pmatrix} \\
L^{(2)} = \begin{pmatrix}
0 & 2 & -2 \\
3 & 0 & -4 \\
1 & 3 & -1
\end{pmatrix}
\]

STOP, there exists a circuit of negative length.
Exercise

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton
Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Exercise

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton
Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Exercise

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton
Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Exercise

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton
Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Exercise

More About Graphs
Tran Vinh Tan

Contents
Connectivity
Paths and Circuits
Euler and Hamilton
Paths
Euler Paths and Circuits
Hamilton Paths and Circuits
Shortest Path Problem
Dijkstra’s Algorithm
Bellman-Ford Algorithm
Floyd-Warshall Algorithm
Traveling Salesman Problem
Planar Graphs
Graph Coloring

Exercise
Application

Problem

A young professor in Hue is invited to teach some years in Ho Chi Minh university of technology. He decides to represent the diverse operations of his transfer by a graph and, in this purpose, establishes the list of following operations:

A: Find a house in Ho Chi Minh city.
B: Choose a removal man and sign a contract of move
C: Make pack his furniture by the removal man
D: Make transport his furniture towards Ho Chi Minh city
E: Find an accommodation to HCM (from Hue)
F: Transport his family to HCM
G: Move into his new accommodation
H: Register the children to their new school
I: Look for a temporary work for his wife
J: Fit out the new accommodation and pay this arrangement with the first treatment of his wife
K: Find a small bar to celebrate in family the success of the move and express the enjoyment to live in a good accommodation arrangement
Application

Considering constraint of posteriority following: $A < F; B < C; C < D \land F; D < G; E < F; F < G \land H \land I; G < K; H < K; I < J; J < K$.

Approximated job processing times:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>
Application

Considering constraint of posteriority following: \( A < F; B < C; C < D \land F; D < G; E < F; F < G \land H \land I; G < K; H < K; I < J; J < K \).

Approximated job processing times:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

Question

- Determine a schedule of the ’movement’ with minimal duration.
- What happens if his new accommodation is not available before date 20? In that case, of what margin we have to make the task \( J \)?
Question

How to determine a shortest path from $u$ to $v$ in graph $G$ which traverses at most $\leq$ a given constant number of intermediate vertices.
Traveling Salesman Problem

Problem

- Given a set of \( n \) customers located in \( n \) cities and distances for each pair of cities, the problem involves finding a round-trip with the minimum traveling cost.
- The vehicle must visit each customer exactly once and return to its point of origin also called depot.
- The objective function is the total cost of the tour.
- \( \mathcal{NP} \)-complete: all known techniques for obtaining an exact solution require an exponentially increasing number of steps (computing resources) as the problems become larger.
- **TSP** is one of the most intensely studied problems in computational mathematics, yet no effective solution method.
The total number of possible Hamilton circuit is $(n - 1)!/2$.

For example, if there are 25 customers to visit, the total number of solutions is $24!/2 = 3.1 \times 10^{23}$.
The total number of possible Hamilton circuit is \((n - 1)!/2\).

For example, if there are 25 customers to visit, the total number of solutions is 
\[
\frac{24!}{2} = 3.1 \times 10^{23}.
\]

If the depot is located at node 1, then the optimal tour is 
\[
1 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1
\]
with total cost equal to 11.
Vehicle Routing Problem

Problem

- The vehicle routing problem involves finding a set of trips, one for each vehicle, to deliver known quantities of goods to a set of customers.
- The objective is to minimize the travel costs of all trips combined.
- There may be upper bounds on the total load of each vehicle and the total duration of its trip.
- The most basic Vehicle Routing Problem (VRP) is the single-depot capacitate VRP.
Planar Graphs

+5V  0V  signal
Planar Graphs
Planar Graphs

**Definition**

- A graph is called **planar** (*phẳng*) if it can be drawn in the plane **without any edges crossing**.
- Such a drawing is called **planar representation** (*biểu diễn phẳng*) of the graph.

\[
K_4 \quad \text{and} \quad K_4 \text{ with no crossing}
\]
Important Corollaries

Corollary

- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, then $e \leq 3v - 6$. 
Important Corollaries

**Corollary**

- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, then $e \leq 3v - 6$.
- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, and no circuits of length 3, then $e \leq 2v - 4$.

$K_{3,3}$
Important Corollaries

Corollary

- If \( G \) is a connected planar simple graph with \( e \) edges and \( v \) vertices where \( v \geq 3 \), then \( e \leq 3v - 6 \).
- If \( G \) is a connected planar simple graph with \( e \) edges and \( v \) vertices where \( v \geq 3 \), and no circuits of length 3, then \( e \leq 2v - 4 \).

\( K_{3,3} \)
Non-planar
Important Corollaries

**Corollary**

- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, then $e \leq 3v - 6$.
- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, and no circuits of length 3, then $e \leq 2v - 4$.
Important Corollaries

Corollary

- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, then $e \leq 3v - 6$.

- If $G$ is a connected planar simple graph with $e$ edges and $v$ vertices where $v \geq 3$, and no circuits of length 3, then $e \leq 2v - 4$.

$K_{3,3}$
Non-planar

$K_5$
Non-planar
Elementary Subdivision

Definition

- Given a planar graph $G$, an elementary subdivision (phan chia sơ cấp) is removing an edge $\{u, v\}$ and adding a new vertex $w$ together with edges $\{u, w\}$ and $\{w, v\}$.
- Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called homeomorphic (đồng phôi) if they can obtained from the same graph by a sequence of elementary subdivisions.
Kuratowski’s Theorem

**Theorem**

A graph is nonplanar iff it contains a subgraph homeomorphic to $K_{3,3}$ or $K_5$. 
Kuratowski’s Theorem

**Theorem**

A graph is nonplanar iff it contains a **subgraph homeomorphic to** $K_{3,3}$ or $K_5$.
Kuratowski’s Theorem

**Theorem**

A graph is nonplanar iff it contains a subgraph homeomorphic to $K_{3,3}$ or $K_5$. 
Kuratowski’s Theorem

Theorem

A graph is nonplanar iff it contains a subgraph homeomorphic to $K_{3,3}$ or $K_5$. 
Exercise

- Is $K_4$ planar?
- Is $Q_3$ planar?
Maps and Graphs

Definition

- Every map can be represented by a graph. We call it dual graph.
- Problem of coloring the regions of a map $\rightarrow$ coloring the vertices of the dual graph so that no two adjacent vertices have the same color.
Maps and Graphs

Definition

• Every map can be represented by a graph. We call it dual graph.
• Problem of coloring the regions of a map → coloring the vertices of the dual graph so that no two adjacent vertices have the same color.
Graph coloring

Definition

- A coloring (tô màu) of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.
# Graph coloring

**Definition**

- **A coloring** (*tô màu*) of a simple graph is the assignment of a color to each vertex of the graph so that no **two adjacent vertices** are assigned the same color.

- **The chromatic number** (*số màu*) of a graph, denoted by \(\chi(G)\), is the least number of colors needed for a coloring of this graph.

```plaintext
Definition

- A coloring (*tô màu*) of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

- The chromatic number (*số màu*) of a graph, denoted by \(\chi(G)\), is the least number of colors needed for a coloring of this graph.

![Diagram of a graph with colored vertices]
```
Graph coloring

Definition

- A coloring (tô màu) of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.
- The chromatic number (số màu) of a graph, denoted by $\chi(G)$, is the least number of colors needed for a coloring of this graph.
Four color theorem

Theorem (Four color theorem)
The chromatic number of a planar graph is no greater than four.

- Was a conjecture in the 1850s
- Was not proved completely until 1976 by Kenneth Appel and Wolfgang Haken, using computer
- No proof not relying on a computer has yet been found
Applications of Graph coloring

Scheduling Final Exam

- How can the final exams at a university be scheduled so that no student has two exams at the same time?
- Suppose we have 7 finals, numbered 1 through 7.
- The pairs of courses have common students are depicted in the following graph

![Graph Diagram]

• How can the final exams at a university be scheduled so that no student has two exams at the same time?
• Suppose we have 7 finals, numbered 1 through 7.
• The pairs of courses have common students are depicted in the following graph
Applications of Graph coloring

**Scheduling Final Exam**

- How can the final exams at a university be scheduled so that no student has two exams at the same time?
- Suppose we have 7 finals, numbered 1 through 7.
- The pairs of courses have common students are depicted in the following graph.
Applications of Graph Coloring

Other Applications

- **Frequency Assignments**: Television channels 2 through 12 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel. How can the assignment of channels be modeled by graph coloring?
Applications of Graph Coloring

Other Applications

- **Frequency Assignments**: Television channels 2 through 12 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel. How can the assignment of channels be modeled by graph coloring?

- **Index Registers**: In an execution of loop, the frequently used variables should be stored in index registers to speed up. How many index registers are needed?