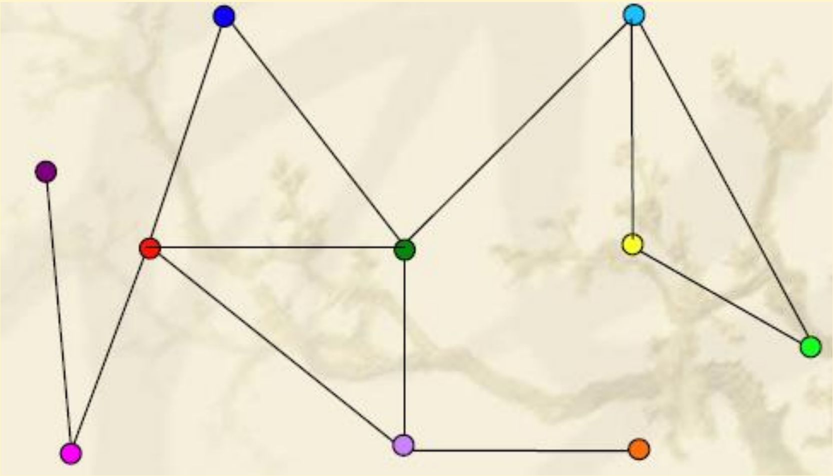
# Modeling of Complex Networks

## Lecture 2: Introduction to Graph Theory

## S8101003Q (Sem A, Fall 2023)

# Review

- Walk: A finite sequence of edges, one after another, in the form of $v_1 v_2$ , $v_2 v_3$ ,..., $v_{n-1} v_n$ where $N(G) = \{v_1,v_2,\ldots,v_n\}$ are nodes.
- A walk is denoted by $v_1 \rightarrow v_2 \rightarrow \ldots v_n$ and the number of edges in a walk is called its length.
- Trail: A walk in which all edges are distinct.
- Path: A trail in which all nodes are distinct, except perhaps $v_1 = v_n$ which is called a closed path, often called a circuit (or, sometimes, a loop or a cycle).
- Tree: A graph with no circuits.

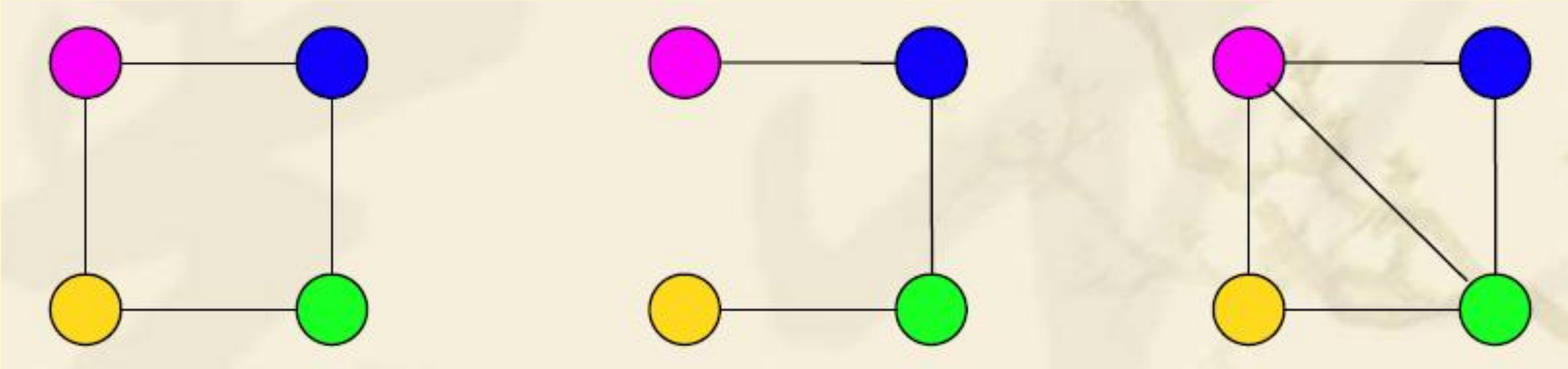


# Eülerian Graphs

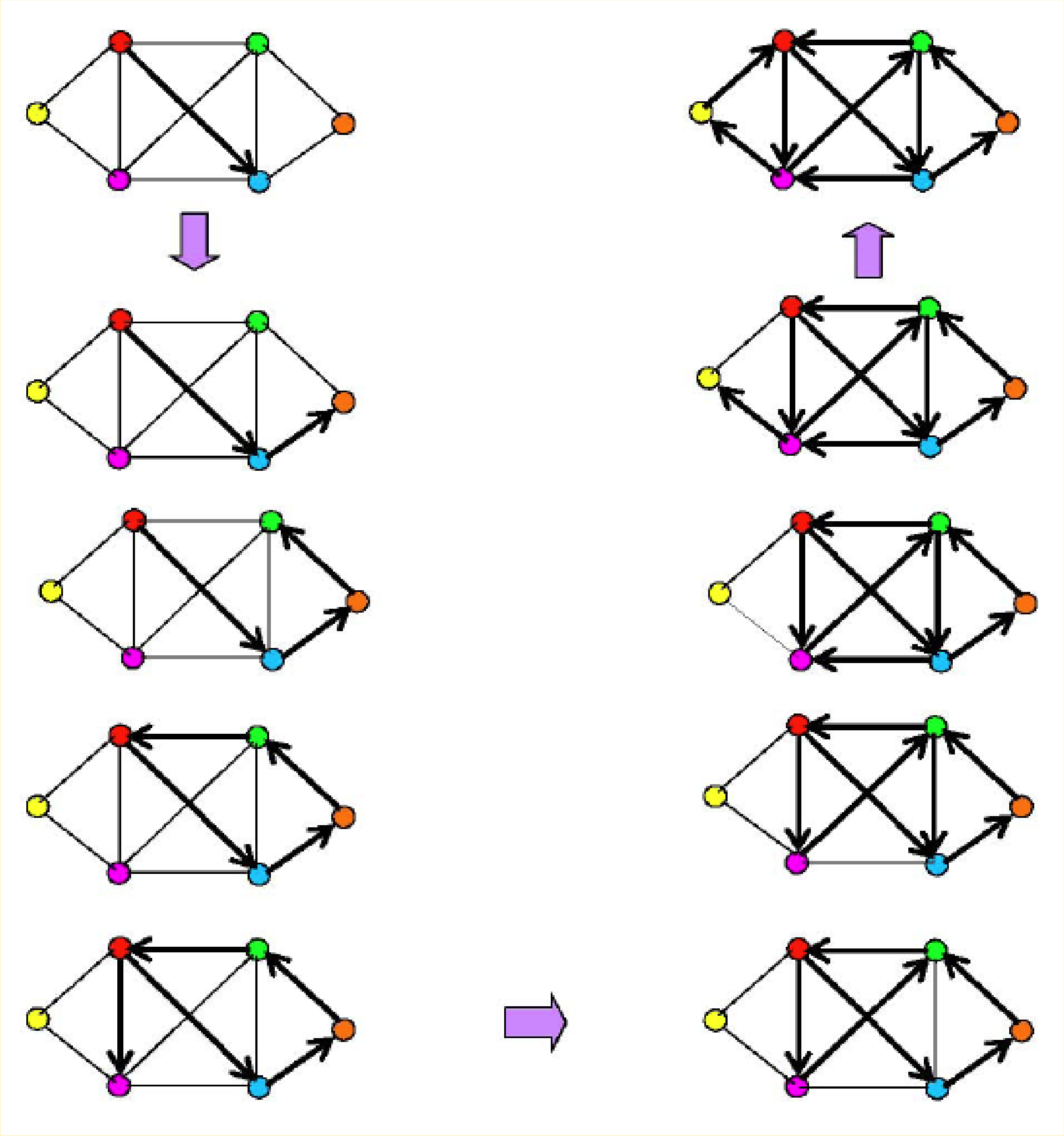
- **Eülerian Graphs:** If it has a closed trail that traverses each edge once and once only.

- There may be more than one such trail, each of which is called an **Eülerian trail**.

- **Semi-Eülerian graph:** If it has a trail, need not be closed, that traverses each edge once and once only.

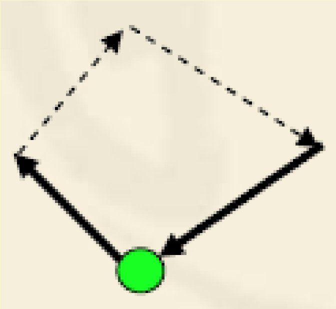


# Example

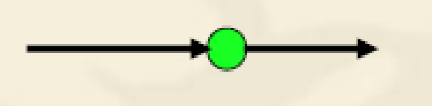


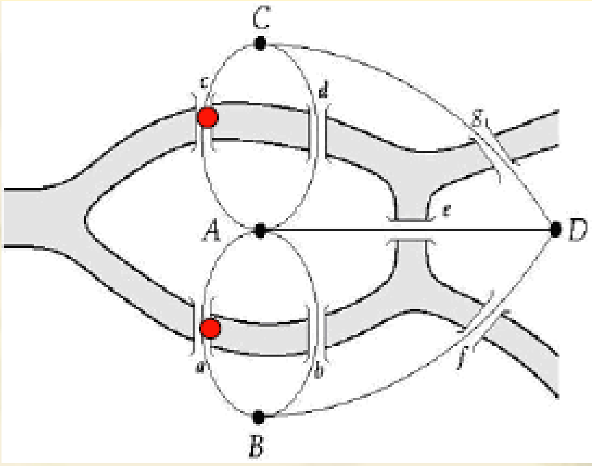
# Main Results

- **Theorem 2-14** *A connected graph is Eülerian if and only if the degree of every node in the graph is an even number.*
- *Proof:* If a node is a starting point, then the trail has to return to it:



If a node has a trail entering it, then the trail has to leave it:





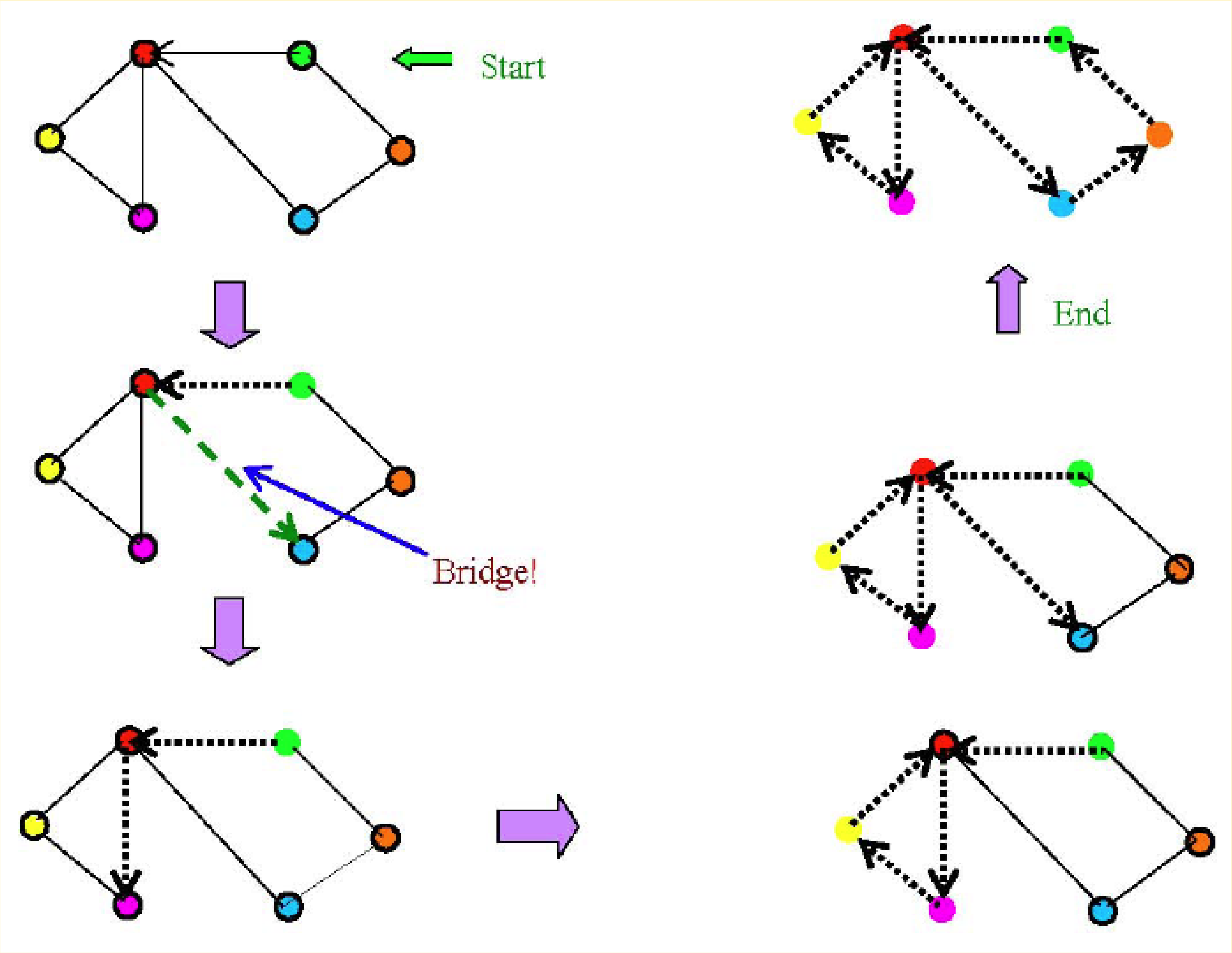

- **Corollary 2-15** *The seven-bridge problem of Königsburg has no solutions.*

# Algorithm to find Eülerian Graphs

- **Corollary 2-16** (Fleury Algorithm) *In any given Eülerian graph* $G$*, an Eülerian trail can be found by the following procedure:*

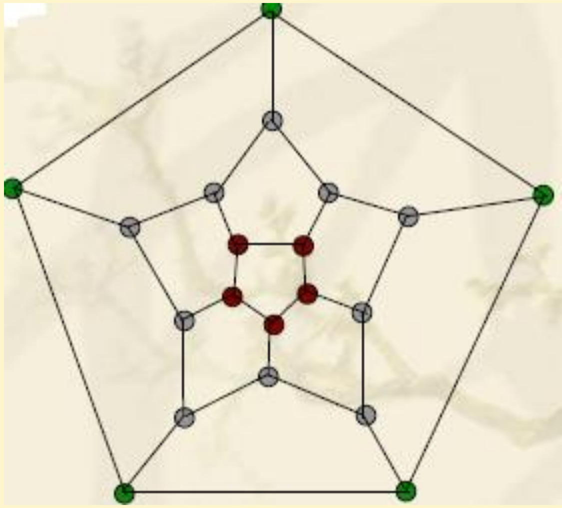
- *Start from any node in* $G$*. Walk along the edges of* $G$ *in an arbitrary manner, subject to the following rules:*
  - *erase the edges as they are traversed;*
  - *erase the resulting isolated nodes;*
  - *walk through a bridge only if there are no other alternatives.*

# Example

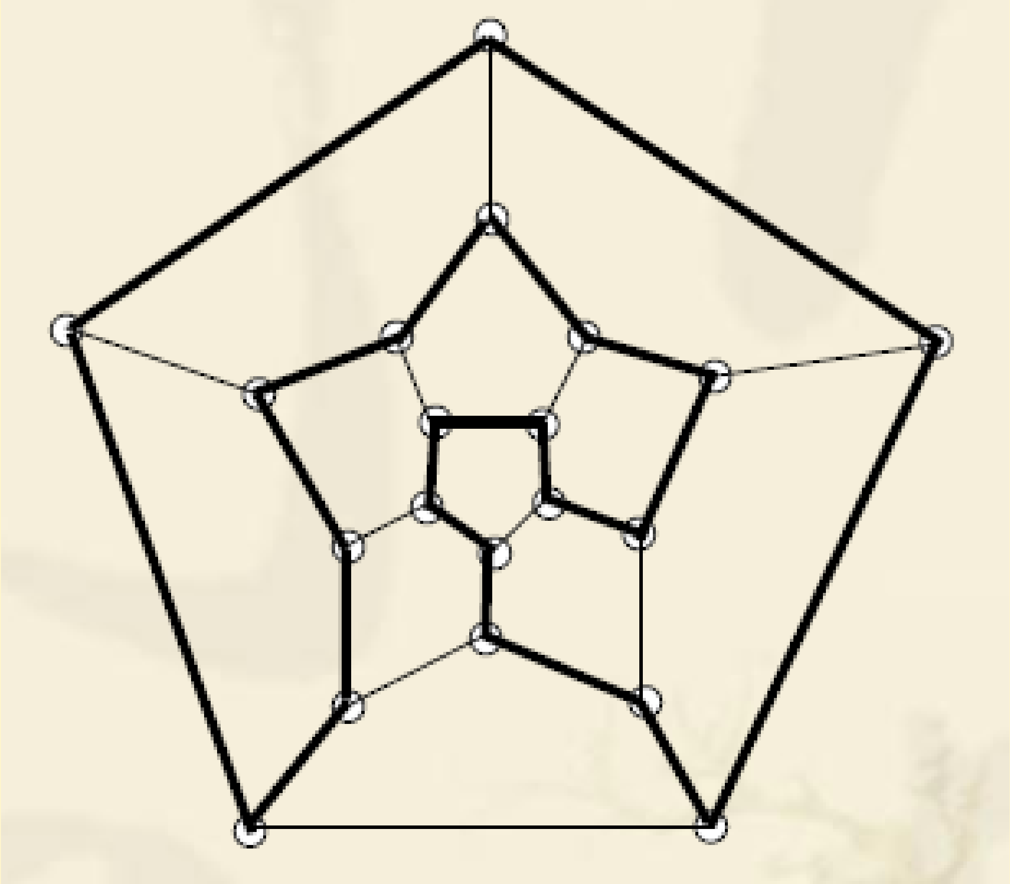


# Sir William Rowan Hamilton (1805-1865)

In 1856, the English mathematician William R. Hamilton studied the world navigation problem and considered a map with 20 nodes representing cities connected by sailing routes, as depicted by the following figure. He wanted to find out if one can traverse through every city once and once only, and finally return to the starting city. His study eventually led to the establishment of the now- famous Hamiltonian graph theory.

# One Solution ?

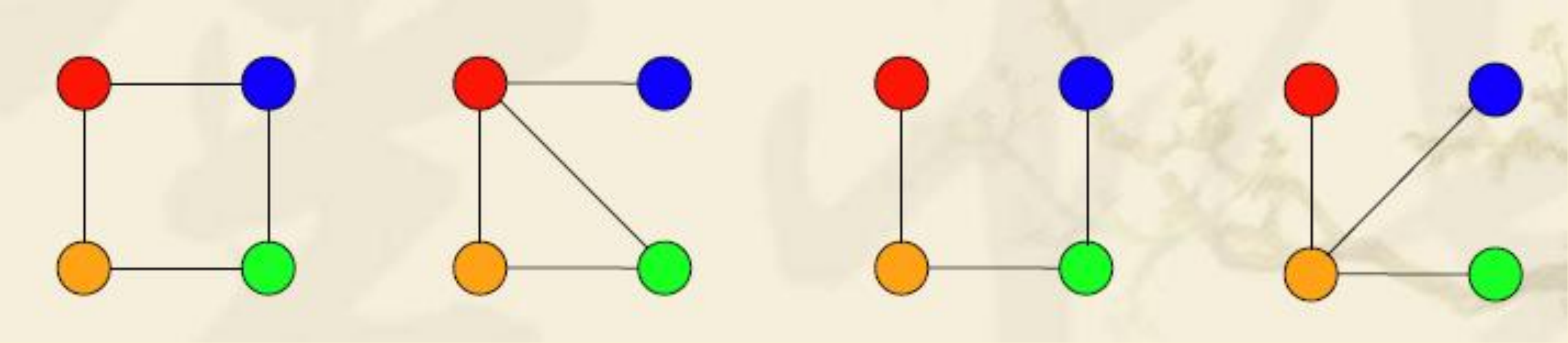


# Hamiltonian Graphs

- **Hamiltonian Graph:** If it has a closed trail that traverses each node once and once only.

- A trivial case is a single node and a nontrivial Hamiltonian graph must be a circuit, called a **Hamiltonian circuit**.

- **Semi-Hamiltonian graph:** If it has a trail, need not be closed, that traverses each node once and once only.



# Some Results

**Theorem 2-17** *Let $G$ be a simple graph with $N$ $(\geq 3)$ nodes. If, for every pair of non-adjacent nodes $v$ and $u$, their degrees always satisfy $k(v)+k(u)\geq N$, then $G$ is Hamiltonian.*

*Proof. Consider any path in the graph, for example:*

$$v_1 - v_2 - v_3$$

*Since* $k(v_1)+k(v_3)\geq 3$, *either $v_1$ or $v_3$ has another neighbor $v_i$, for example,*

$$\begin{array}{ccccc} & & v_i & & \\ & & | & & \\ v_1 & - & v_2 & - & v_3 \end{array}$$

*But then* $k(v_1)+k(v_i)\geq 3$, *so either $v_1$ or $v_i$ has another neighbor,* **……**

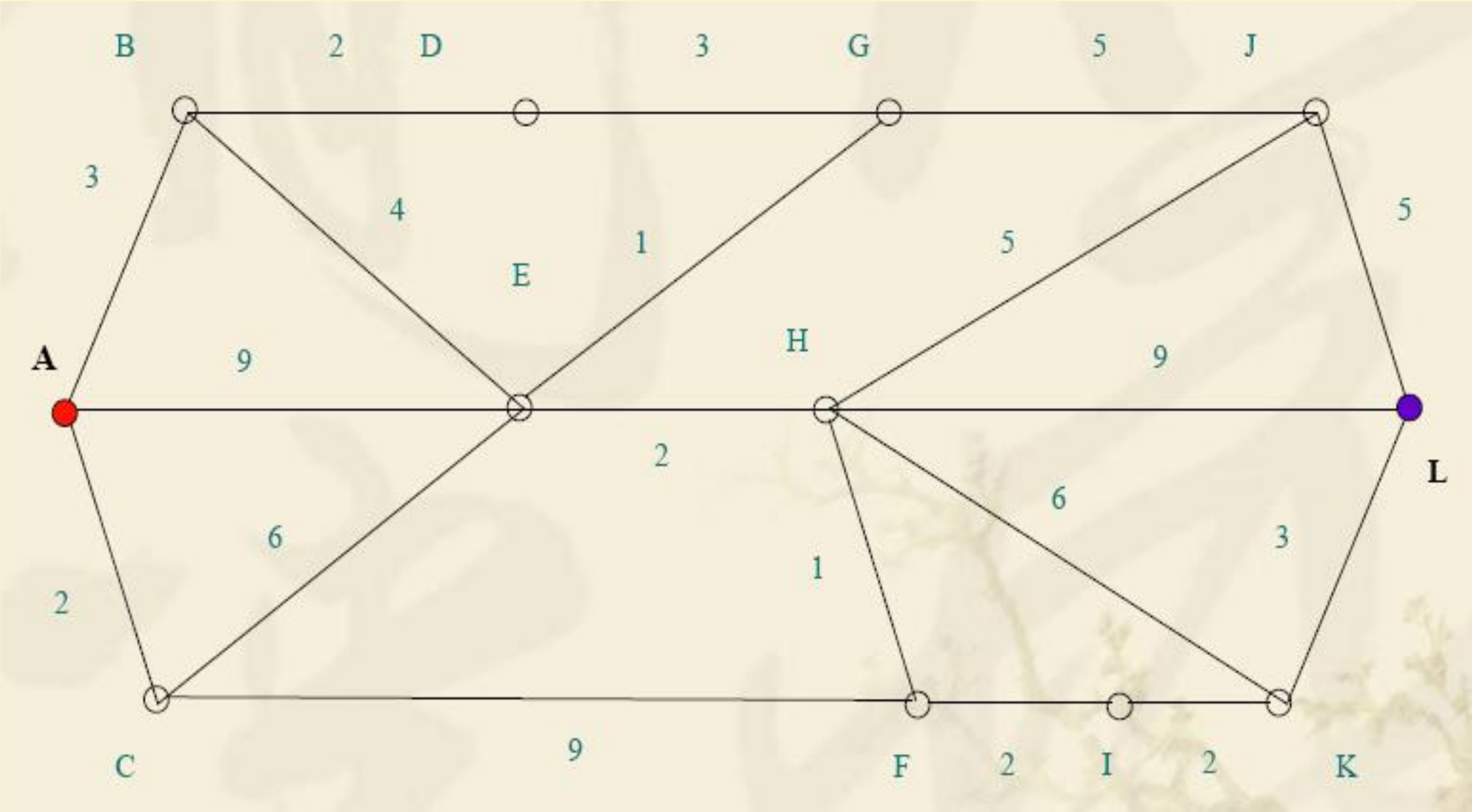
*Thus, the only way to satisfy* $k(v_1)+k(v_i)\geq 3$ *for non-adjacent $v_1$ and $v_i$ is to close the loop:*

$$\begin{array}{ccccc} & -\dots & & & v_i \\ | & & & & | \\ v_1 & - & v_2 & - & v_3 \end{array}$$

*This gives a Hamiltonian circuit.*

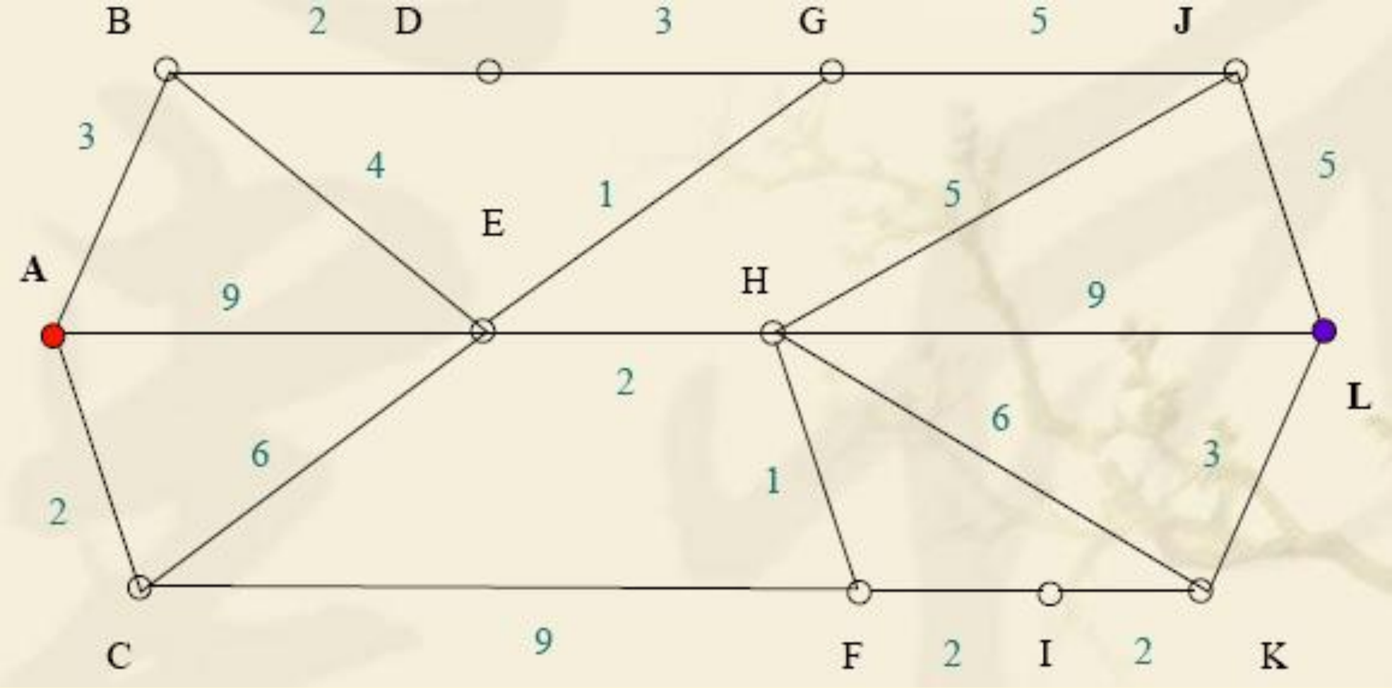
# Shortest Path Length Problem

**Q:** What is the shortest path length from *A to L ?*



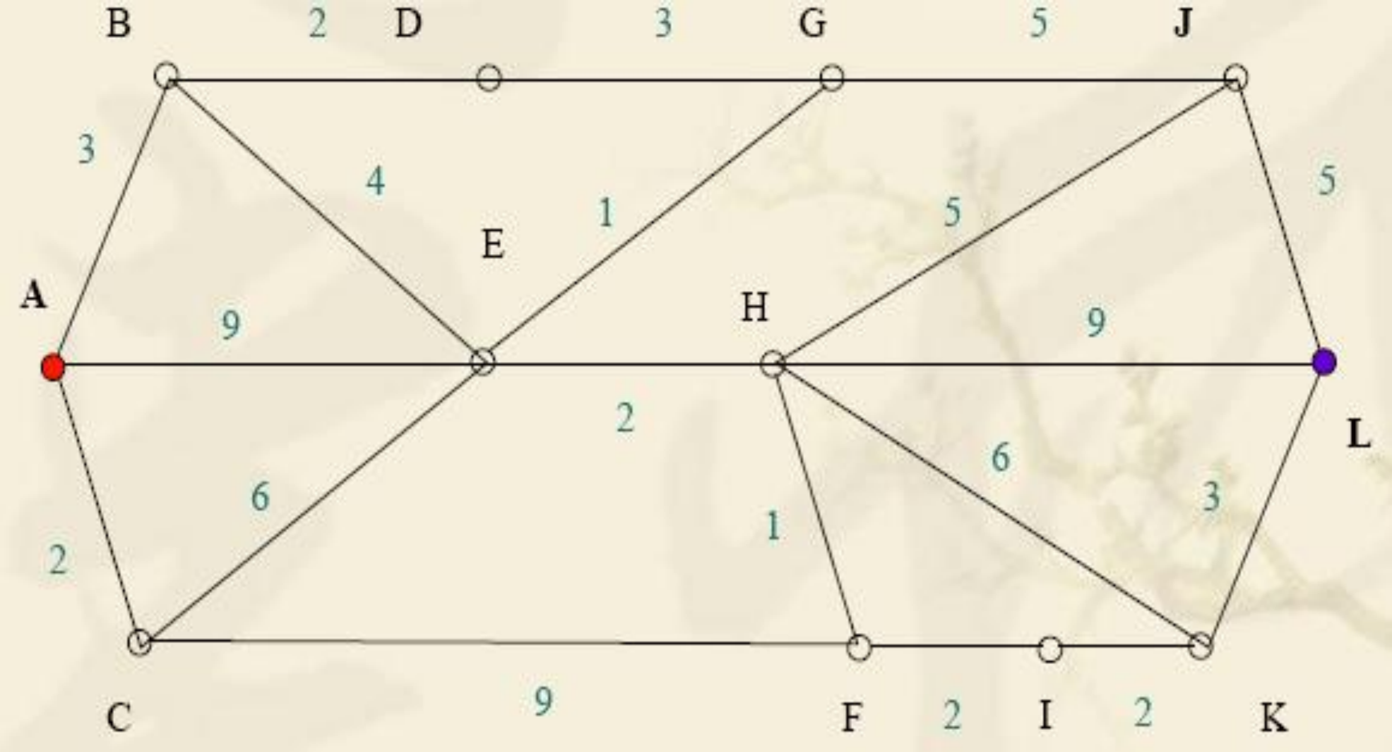
# Approach to Solving the Shortest Path Length Problem

- Moving from $A$ toward $L$ and associating each intermediate node $V$ with a number $l(V)$ that is equal to the shortest path length from $A$ to $V$.
- For example: from $A$ to $J$, one has $l(J)=l(G)+5$ or $l(J)=l(H)+5$ whichever is shorter.



# Approach to Solving the Shortest Path Length Problem

- Starting from $A$ with $l(A)=0$, moving from $A$ toward $L$, one has $l(B)=l(A)+3=3$; $l(E)=L(A)+9=9$; $l(C)=l(A)+2=2$
- Therefore, node $B$ is chosen, with $l(B)=3$, node $C$ is chosen, with $l(C)=2$ and node $E$ is chosen with $l(E)=9$

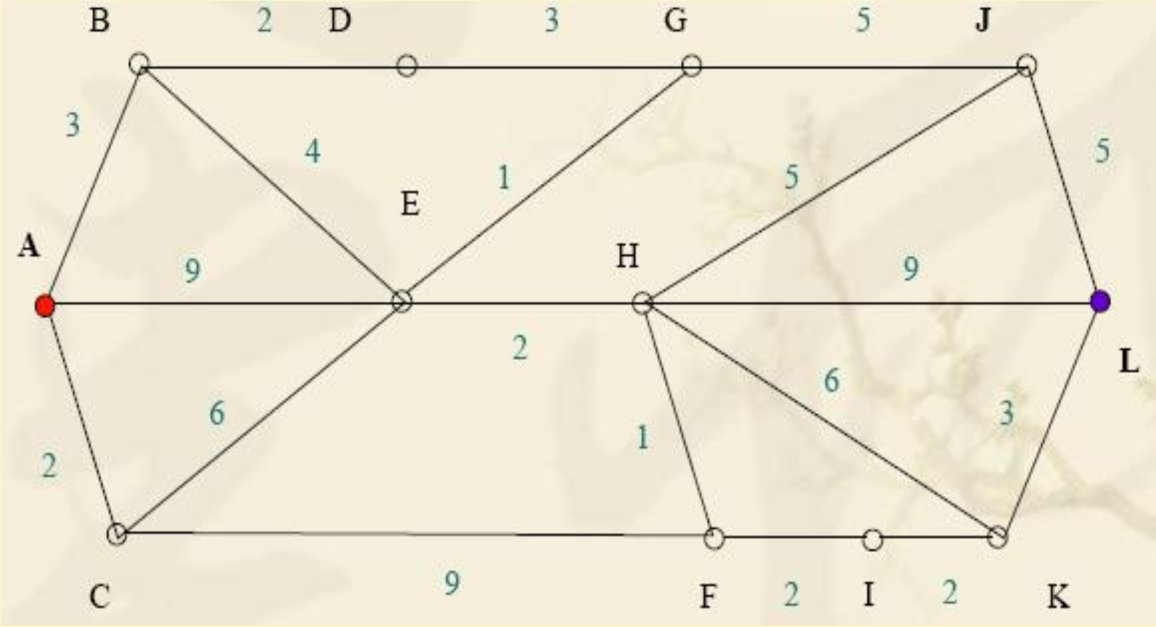


# Approach to Solving the Shortest Path Length Problem

- Looking at the nodes adjacent to $B$ , one has

  $l(D)=l(B)+2=5;\ l(E)=L(B)+4=7$
- Looking at the nodes adjacent to $C$ , one has

  $l(E)=l(C)+6=8;\ l(F)=l(C)+9=11$
- So, $l(D)=5$ is uniquely determined.
- But $l(E)=7$, or $8$ or $9$. Hence, node $E$ is chosen with $l(E)=l(B)+4=7$
- $F$ is chosen with $l(F)=11$
- But also $l(F)=l(H)+1$. Looking at $l(H)=l(E)+2=9$, so $F$ is chosen with $l(F)=l(H)+1=10$



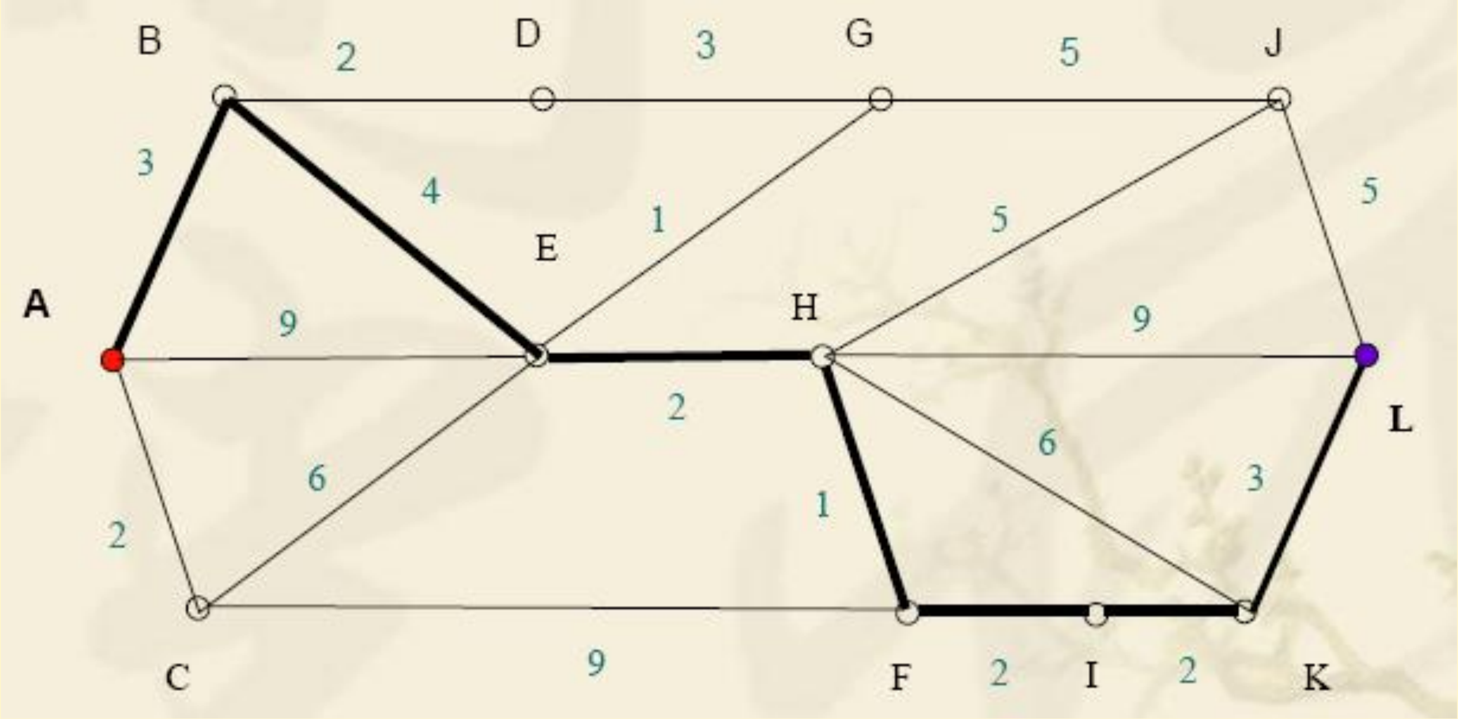
# Approach to Solving the Shortest Path Length Problem

- Continuing this way, one obtains

  $l(A)=0$  $l(B)=3$  $l(C)=2$  $l(D)=5$  $l(E)=7$  $l(F)=10$

  $l(G)=8$  $l(H)=9$  $l(I)=12$  $l(J)=13$  $l(K)=14$  $l(L)=17$

- Final result: total shortest path length $=17$ from $A$ to $L$:




- Solutions may not be unique, but all will be the shortest

# Chinese Postman Problem

- The problem is for a postman to deliver all letters in such a way that he passes every street at least once and finally returns to the starting point (the post office), traversing a shortest possible total path-length.
- It differs from an Eülerian graph in that it allows multiple edges between two nodes. Also, it has a requirement of having minimal path-length.
- First solution given by Mei-ko Kwan (1934—) (管梅谷 from Shanghai, China) in 1962

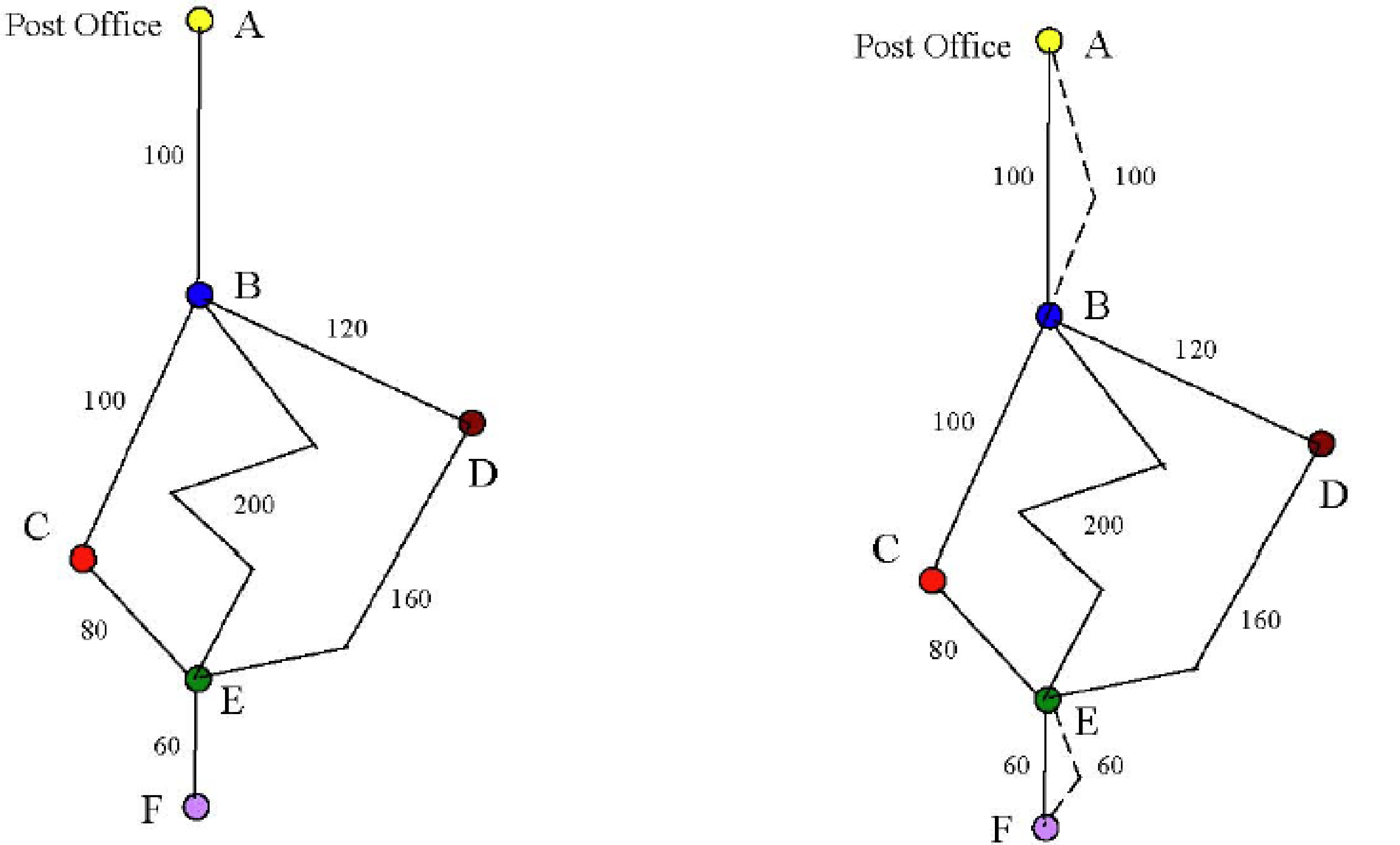
# Chinese Postman Problem

- **Kwan's Algorithm**

  In any given connected graph $G$, a Chinese postman trail can be found by the following off-line procedure:

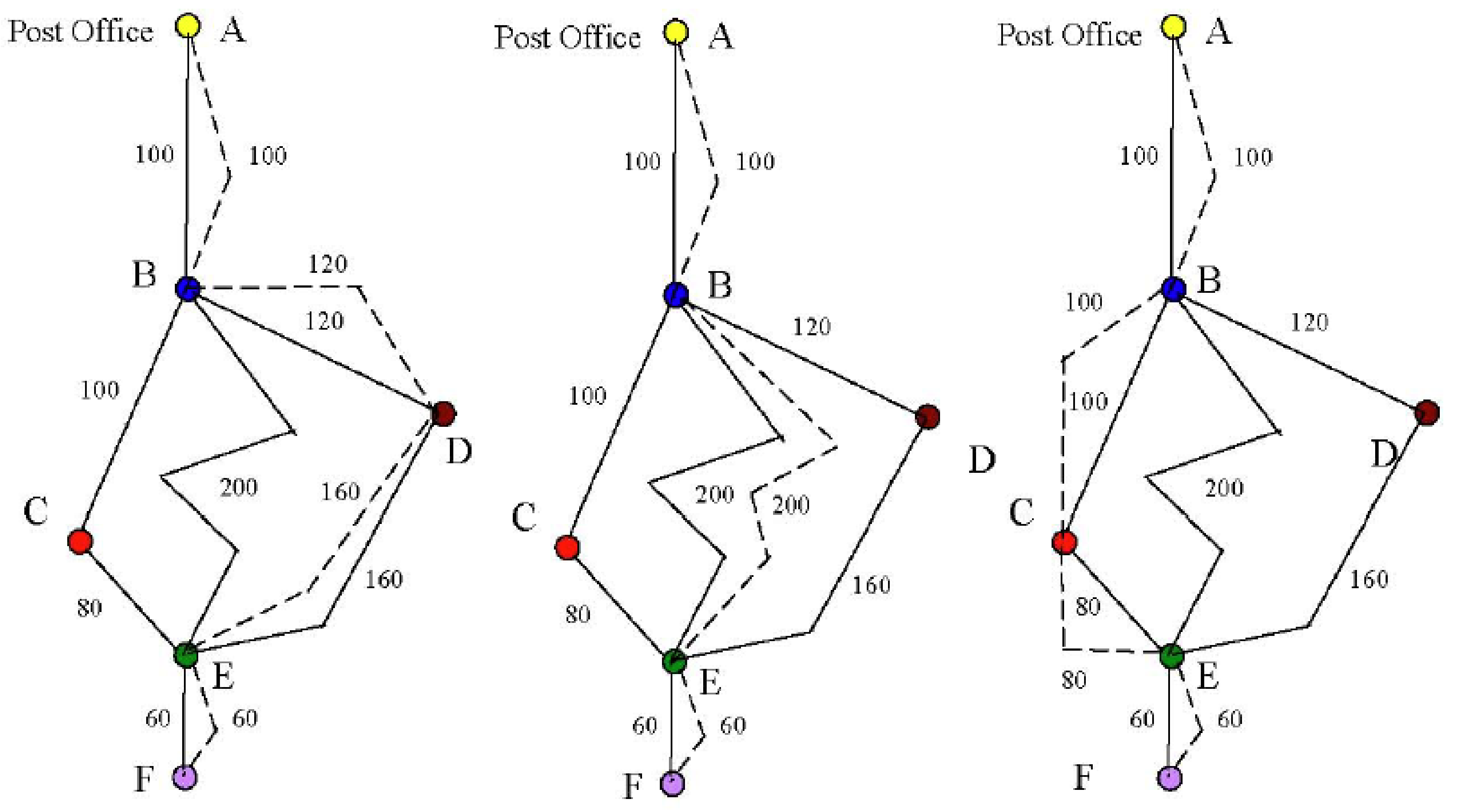
- Identify the initial node in $G$, and then do the following:
- for each odd-degree node in $G$, add an edge to connect it to one of its neighboring node;
- repeat the above, until all nodes in $G$ have even degrees;
- check all resulting reconfigurations -- if the increment of the total path-length of every loop so created is not longer than one half of the total path-length of the corresponding original loop, then keep this loop as a solution.

# Example




(a) The routing map (b) Making nodes A and F be of even degree

# Example




(c) Possible route 1

(d) Possible route 2

(e) Possible route 3

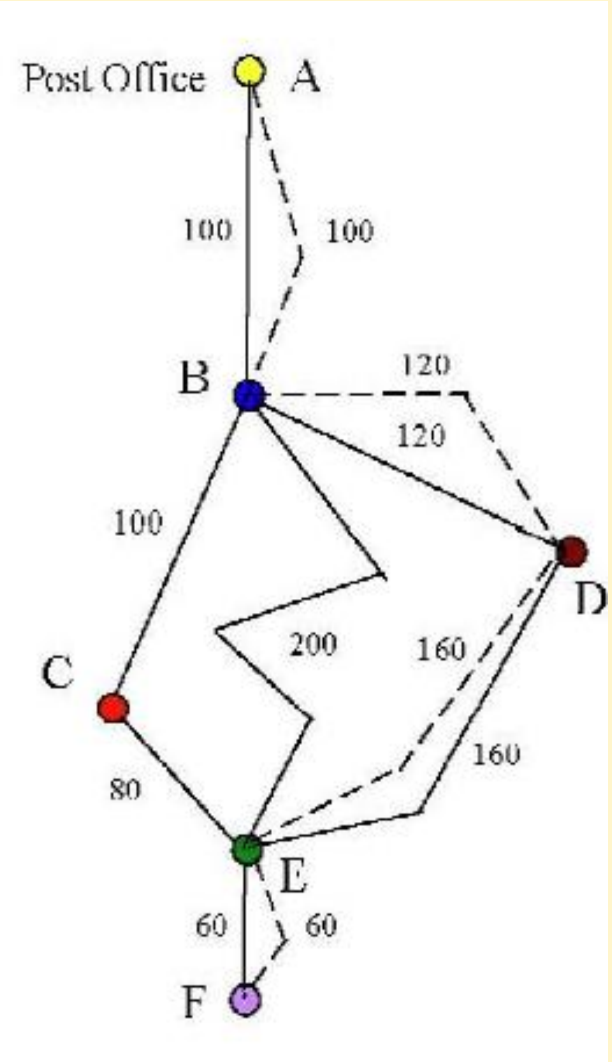
# Example

There are 3 existing loops:

(1)Total path-length of B-D-E-C-B = 120 + 160+ 80 + 100 = 460
(2) Total path-length of B-C-E-B = 100 + 80 + 200 = 380 [this is unrelated with increment]
(3)Total path-length of B-D-E-B = 120 + 160 + 200 = 480

Possible choices of routs are:

Fig. (c): New increment of total path-length = 120 + 160 = 280, which is longer than one half of the corresponding loop B-D-E-C-B = 460/2 = 230, and also longer than one half of the corresponding loop B-D-E-B = 480/2 = 240.
→Not a solution.



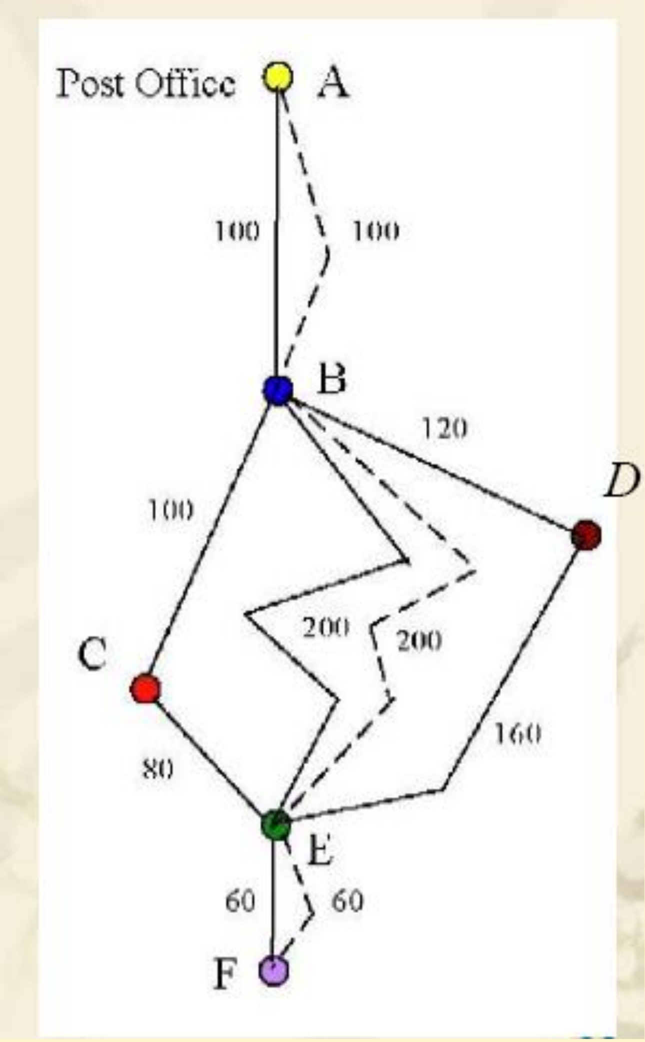
# Example

There are 3 existing loops:

(1)Total path-length of B-D-E-C-B = 120 + 160+ 80 + 100 = 460 [this is unrelated with increment]
(2) Total path-length of B-C-E-B = 100 + 80 + 200 = 380
(3)Total path-length of B-D-E-B = 120 + 160 + 200 = 480

Possible choices of routs are:

Fig. (d): New increment of total path-length = 200, which is longer than one half of the corresponding loop B-C-E-B = 380/2 = 190, although not longer than one half of the corresponding loop B-D-E-B = 480/2 = 240.
→Not a solution.



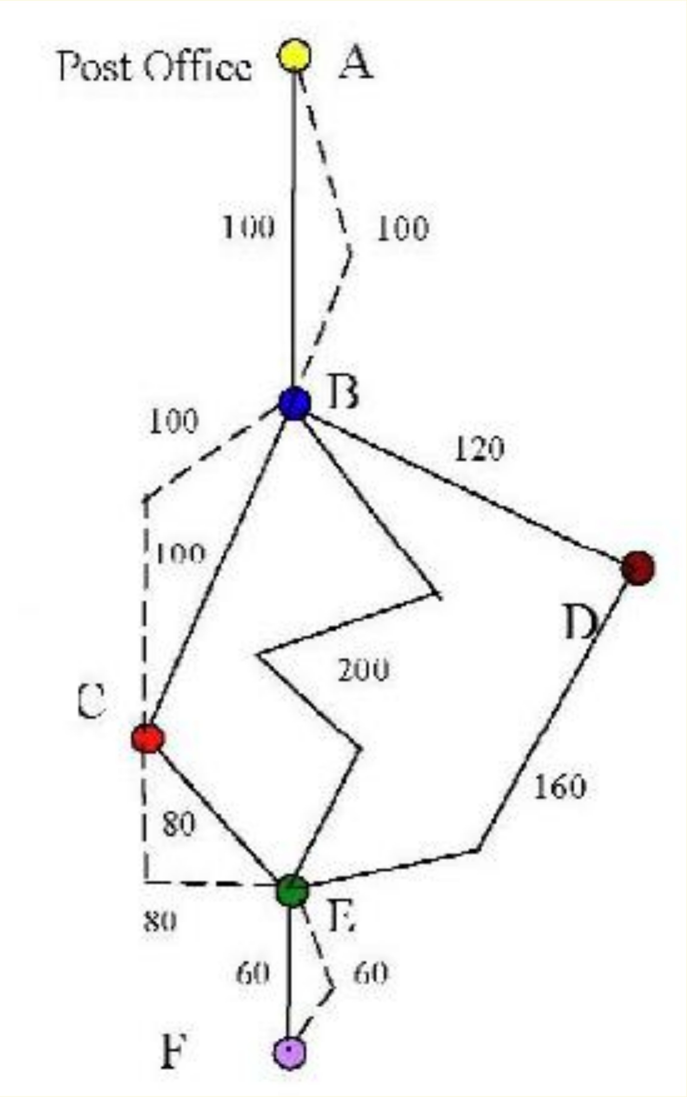
# Example

There are 3 existing loops:

(1)Total path-length of B-D-E-C-B = 120 + 160+ 80 + 100 = 460
(2) Total path-length of B-C-E-B = 100 + 80 + 200 = 380
(3)Total path-length of B-D-E-B = 120 + 160 + 200 = 480 [this is unrelated with increment]

Possible choices of routs are:

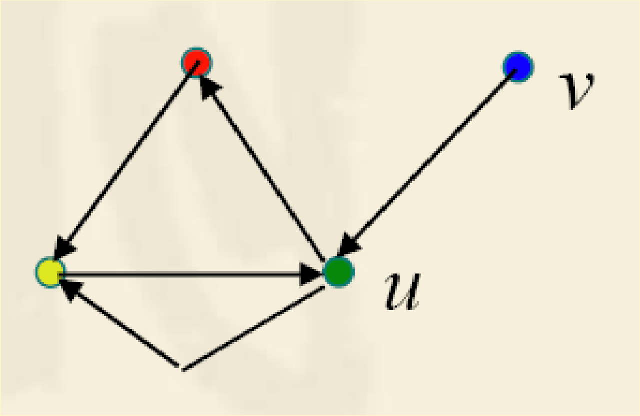
Fig. (e): New increment of total path-length 100+80 = 180, which is shorter than one half of the corresponding loop B-C-E-B = 380/2 = 190, and also shorter than one half of the corresponding loop B-C-E-D-B = 460/2 = 230.
This is a solution → A-B-C-E-F-E-D-B-E-C-B-A.
(solution may not be unique in general)



**The following part is not required; it is for information only**

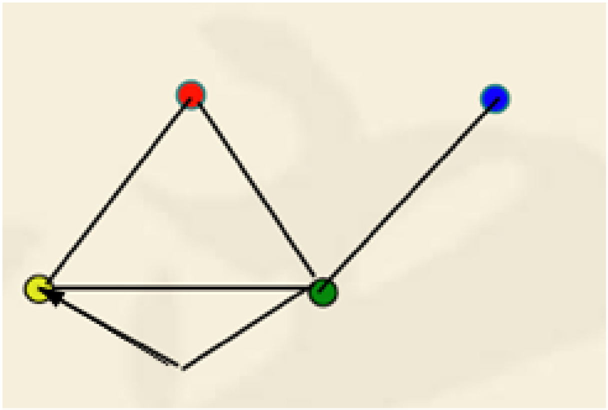
# Directed Graphs

- Digraph (=directed graph), $(N(Gd),E(Gd))$ edge$(u,v)$: $u \leftarrow v$




Simple digraph

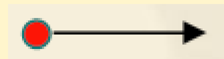
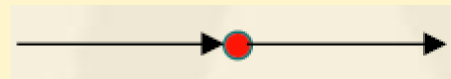
- Underlying graph, $(N(G),E(G))$ is obtained by removing all direction arrows in the original digraph

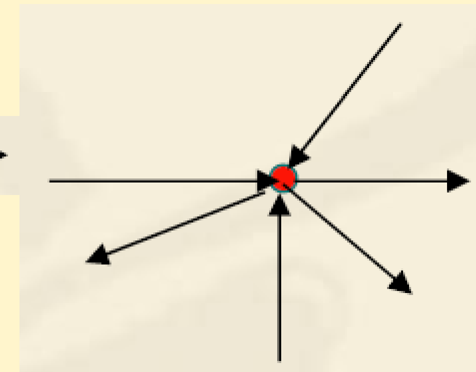


Non-simple graph

- Two digraphs are isomorphic if their underlying graphs are isomorphic

# Degrees

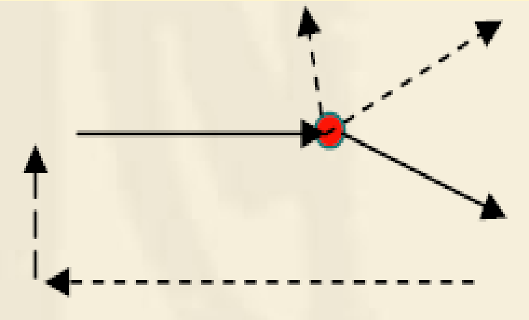
- In-degree: *d_in* 
- Out-degree: *d_out* 
- Balanced-degree: *d=d_in + d_out* 



- The following can be similarly defined:
  - Directed walk
  - Directed trail
  - Directed path
  - Directed circuit
  - Directed tree
  - Directed Eülerian graph
  - Directed Hamiltonian graph

# Some Results

Theorem. Let $G$ be a connected digraph. If all its nodes have $d\_in = 1$, regardless of their $d\_out$, then $G$ has one and only one directed circuit.



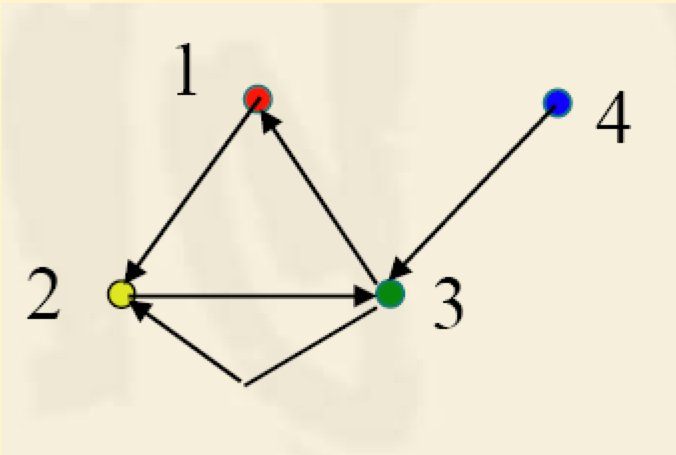
Proof. Start from any node, say $u$. This node has one and only one incoming edge by assumption. Another end node of this incoming edge also has one and only one incoming edge by assumption. Keep moving backward this way. In the backward sequence of edges, since every node has one and only one incoming edges, the sequence will not hit any node of the sequence except possibly the very first one, $u$, and it will eventually do so because the network has finitely many nodes.

Corollary. Let $G$ be a connected digraph. If all its nodes have $d\_out = 1$, regardless of their $d\_in$, then $G$ has one and only one directed circuit.

Theorem. A connected digraph is a directed circuit if and only if all its nodes satisfy $d\_in = d\_out = 1$.

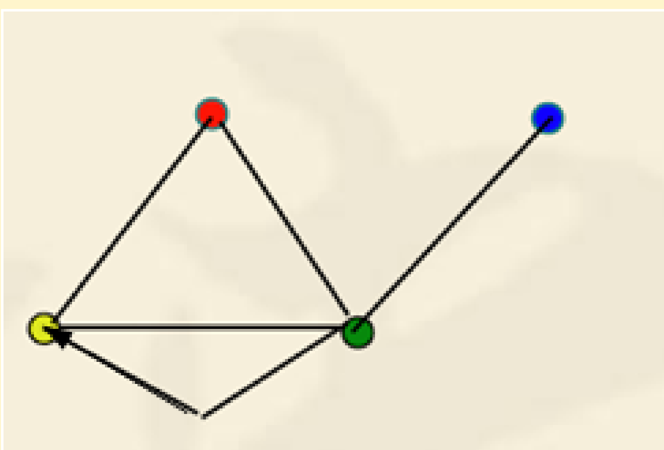
Corollary. A connected digraph contains a directed circuit if and only if it has a subgraph with all nodes satisfying $d\_in = d\_out = 1$.

# Adjacency Matrix




$$A_d = [a_{ij}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$ (asymmetric)

$a_{ij}$ -- from $i$ to $j$

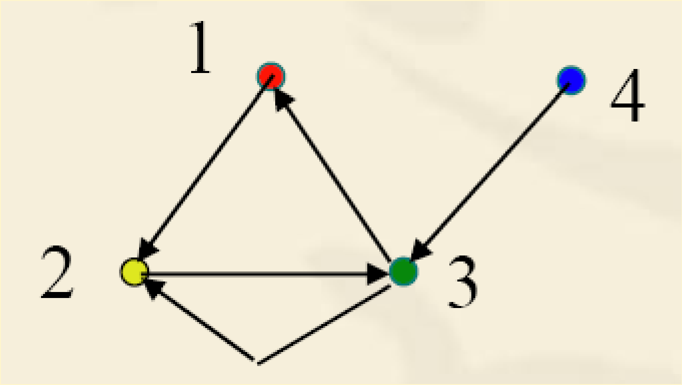


Underlying network
$A$ is not well-defined
due to multiple edges

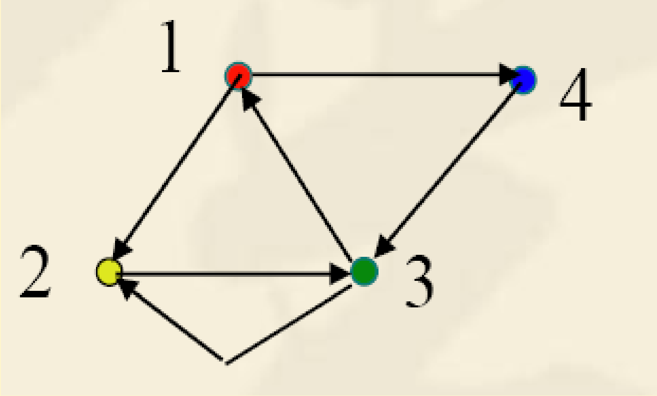
# Connectivity

A digraph is connected if its underlying graph is connected.

A digraph is strongly connected if there is a directed path from any node to any other node.




← connected but not strongly connected




← strongly connected

# Some Results

Theorem. A connected digraph is directed Eulerian if and only if every node satisfies

$$d\_in = d\_out$$

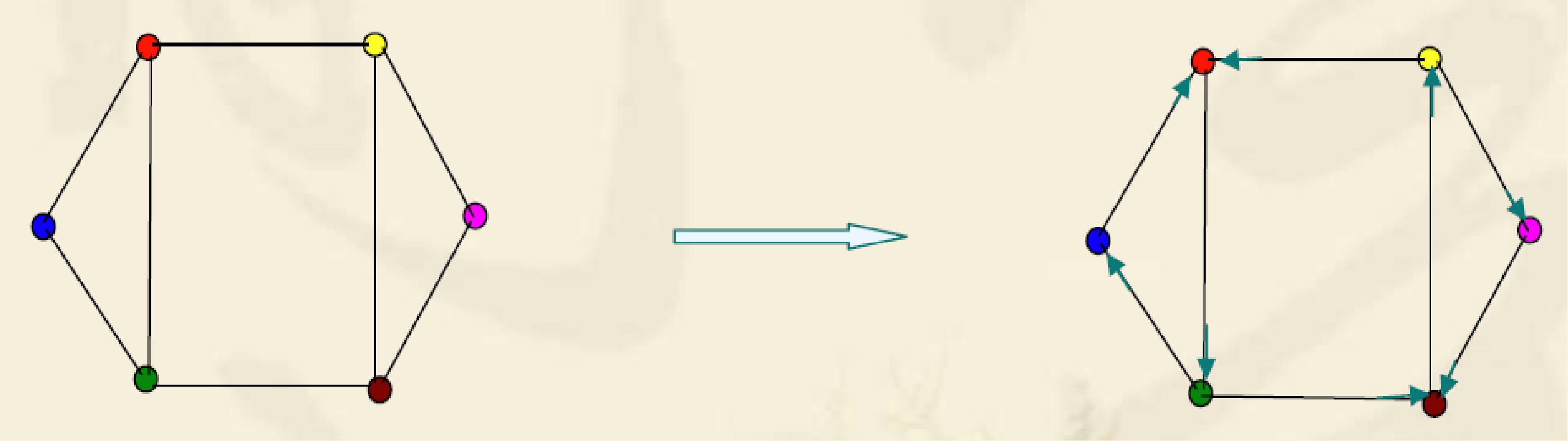
Theorem. Let $G$ be a strongly connected digraph with $N$ nodes. If all nodes satisfy both

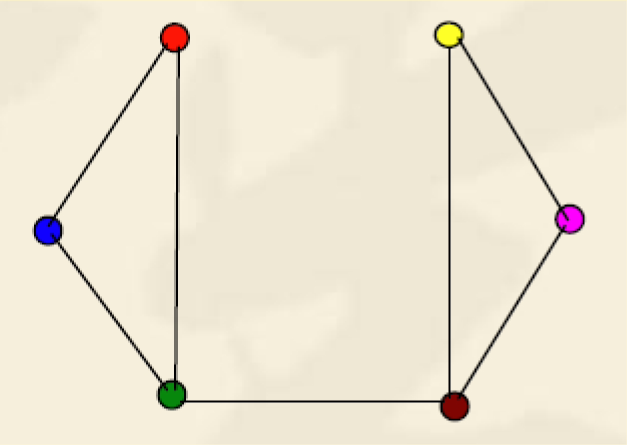
$d_{in} \geq N/2$ and $d_{out} \geq N/2$

Then $G$ is directed Hamiltonian.

# Orientable Graphs

A graph is orientable if its edges can be directed such that the resulting digraph is strongly connected.





← not orientable

→ importance of bridges !

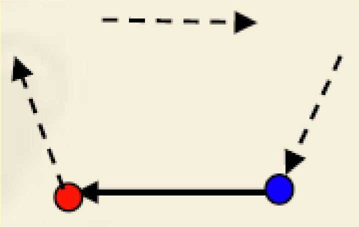
→ importance of circuits !

# Some Results

**Theorem:** A connected graph is orientable if and only if each edge of the graph is contained in a circuit.

*Proof.*

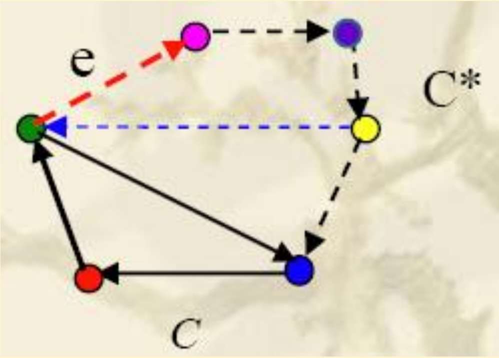
Necessity - $G$ is orientable → every edge can be directed →



Sufficiency - Choose any circuit $C$ and direct it cyclically.

If $C = G$ , done.

If $C \subset G$ , then consider any adjacent edge $e$. By assumption, $e$ is in some circuit $e \in C^* \subset G$ Since $C^*$ is a circuit, there is a path to come back to *e.* One can then direct this path cyclically.

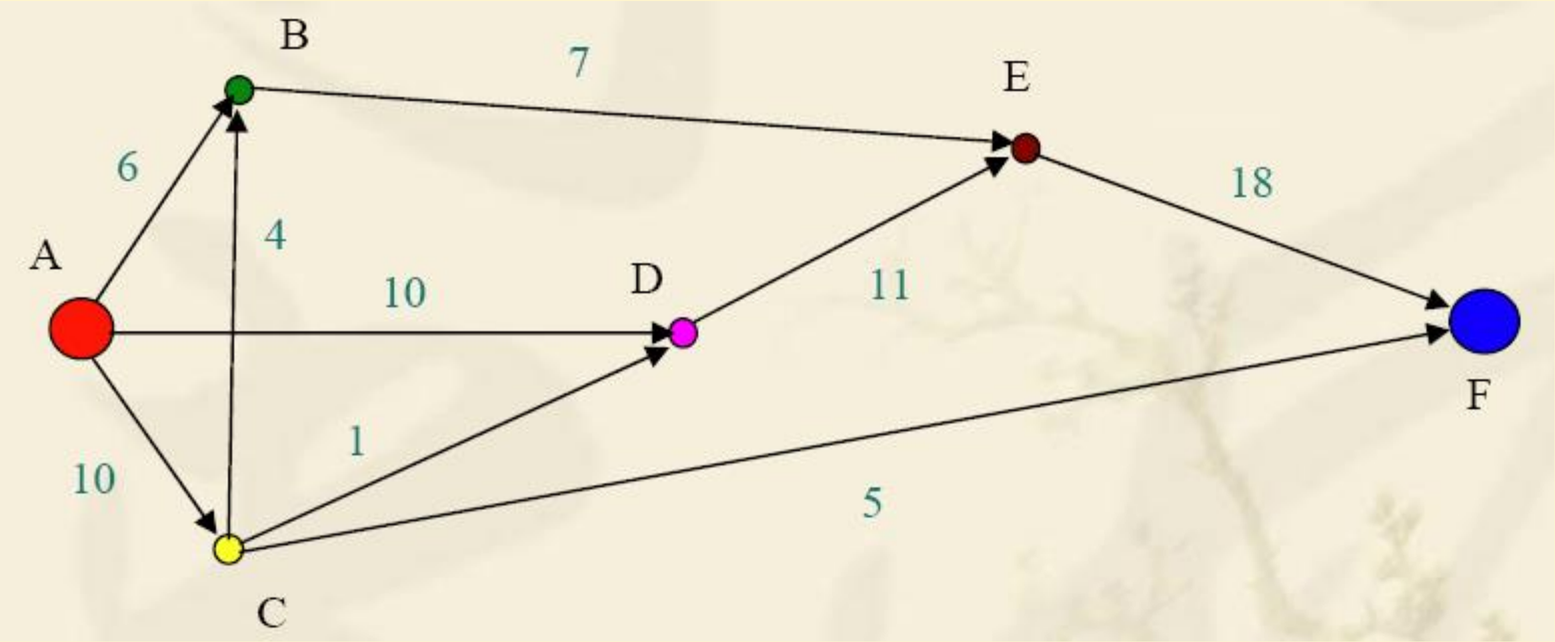



Repeat the above for all edges. Done.

# Application: Maximum Flows

Consider the following roadmap, where weights are profit values.

Starting from $A$, follow the directed road to traverse to $F$, not allowed to repeat any road, finally arrive at $F$ while making the maximum profits.



# Algorithm

(similar to the shortest path length problem)

$l(A) = 0$

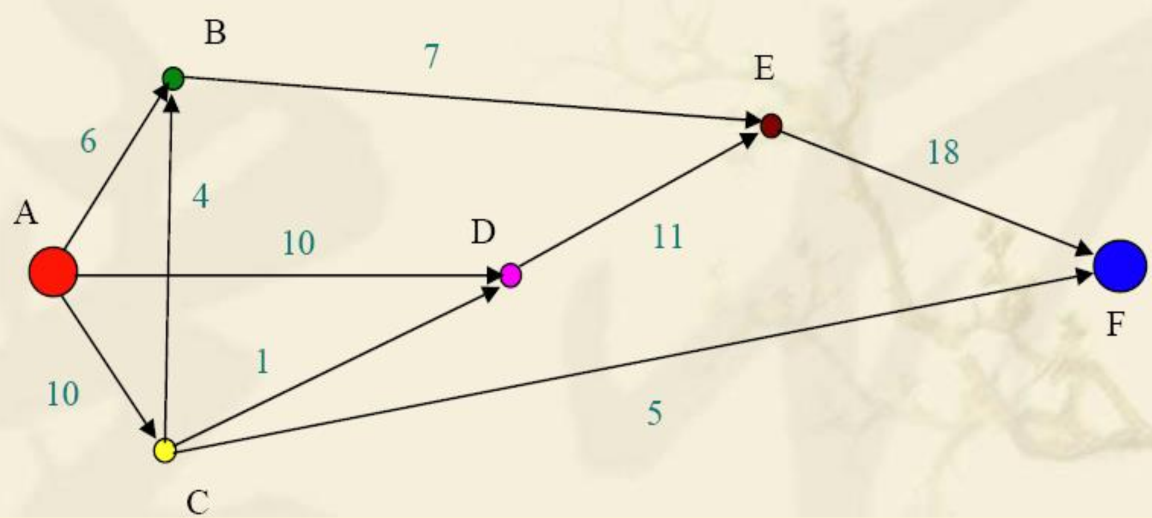
$l(B) = l(A) + 6 = 6$

$l(C) = l(A) + 10 = 10 \rightarrow l(B) = l(C) + 4 = 14 \rightarrow l(E) = l(B) + 7 = 21 \rightarrow l(F) = l(E) + 18 = \mathbf{39}$

$l(D) = l(C) + 1 = 11 \rightarrow l(E) = l(D) + 11 = 22 \rightarrow l(F) = l(E) + 18 = \mathbf{40}$

$l(F) = l(C) + 5 = \mathbf{15}$

$l(D) = l(A) + 10 = 10 \rightarrow l(E) = l(D) + 11 = 21 \rightarrow l(F) = l(E) + 18 = \mathbf{39}$



# Optimal Solution

$l(A) = 0$

$l(B) = l(A) + 6 = 6$

$l(C) = l(A) + 10 = 10 \rightarrow l(B) = l(C) + 4 = 14 \rightarrow l(E) = l(B) + 7 = 21 \rightarrow l(F) = l(E) + 18 = \mathbf{39}$

$l(D) = l(C) + 1 = 11 \rightarrow l(E) = l(D) + 11 = 22 \rightarrow l(F) = l(E) + 18 = \mathbf{40}$

$l(F) = l(C) + 5 = \mathbf{15}$

$l(D) = l(A) + 10 = 10 \rightarrow l(E) = l(D) + 11 = 21 \rightarrow l(F) = l(E) + 18 = \mathbf{39}$