LECTURE # 07, 08 & 09

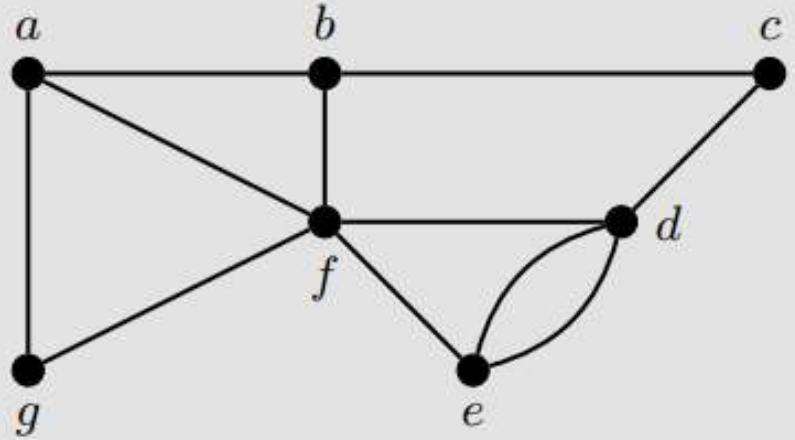
Definition 2.1 Let G be a graph.

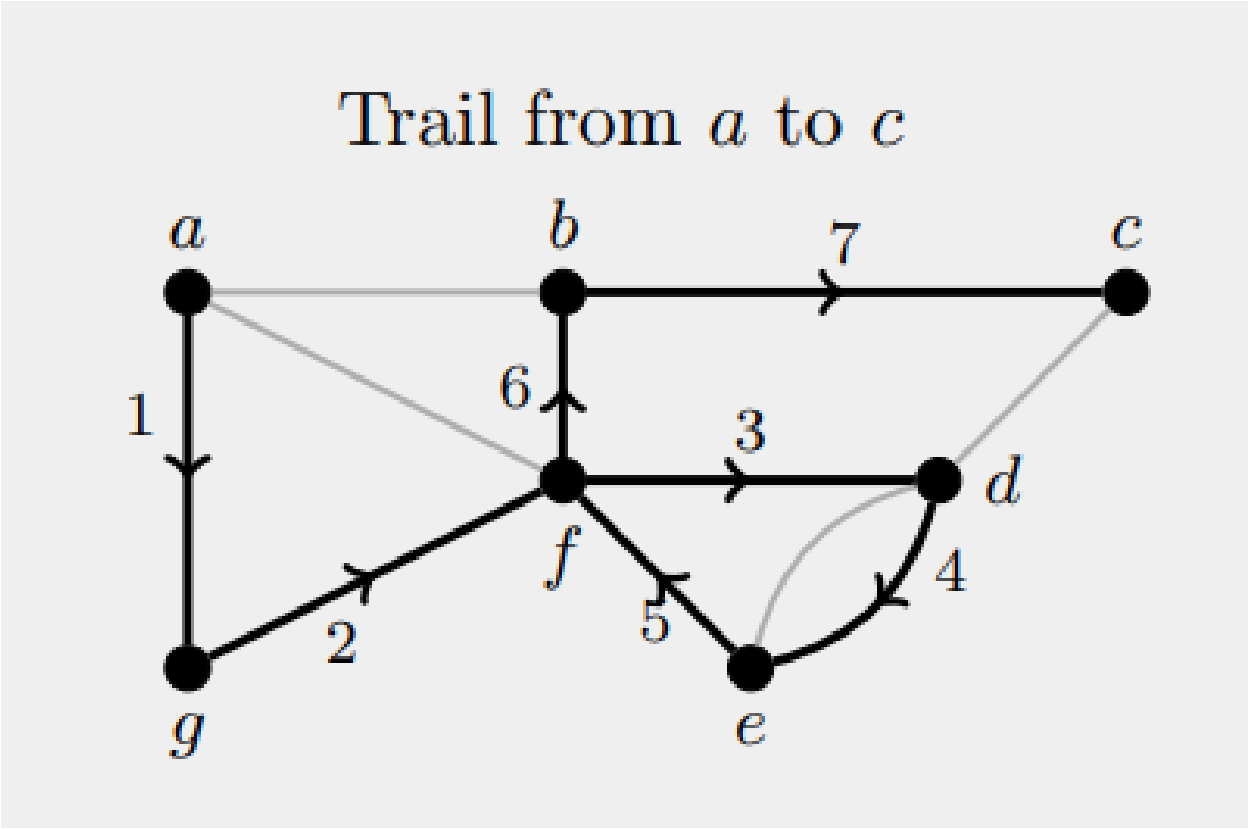
- A ***walk*** is a sequence of vertices so that there is an edge between consecutive vertices. A walk can repeat vertices and edges.

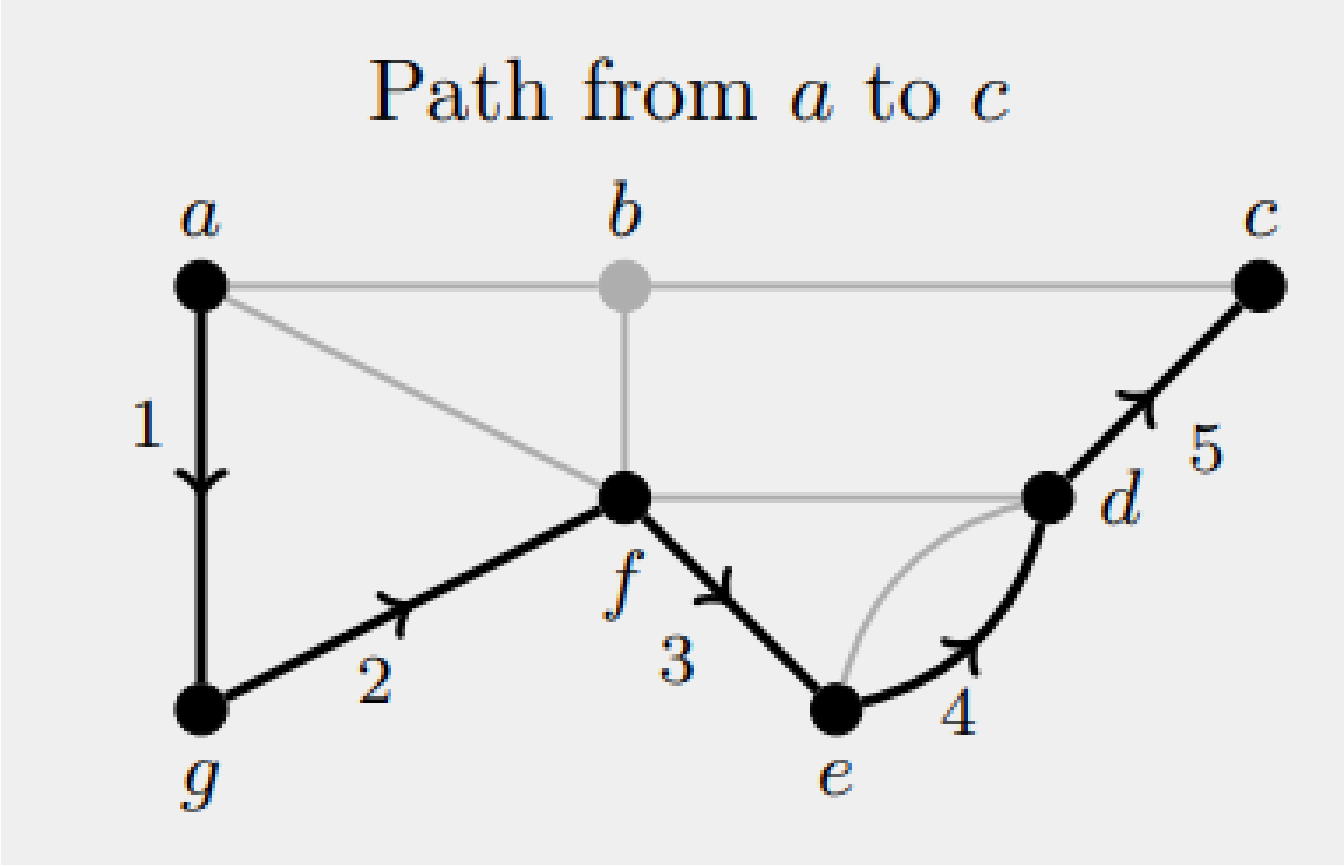
- A ***trail*** is a walk with no repeated edges. A trail can repeat vertices but not edges.
- A ***path*** is a trail with no repeated vertex (or edges). A path on n vertices is denoted P_n.
- A ***closed walk*** is a walk that starts and ends at the same vertex.
- A ***circuit*** is a closed trail; that is, a trail that starts and ends at the same vertex with no repeated edges though vertices may be repeated.
- A ***cycle*** is a closed path; that is, a path that starts and ends at the same vertex. Thus cycles cannot repeat edges or vertices. Note: we do not consider the starting and ending vertex as being repeated since each vertex is entered and exited exactly once. A cycle on n vertices is denoted C_n.

The ***length*** of any of these tours is defined in terms of the number of edges. For example, P_n has length $n-1$ and C_n has length n.

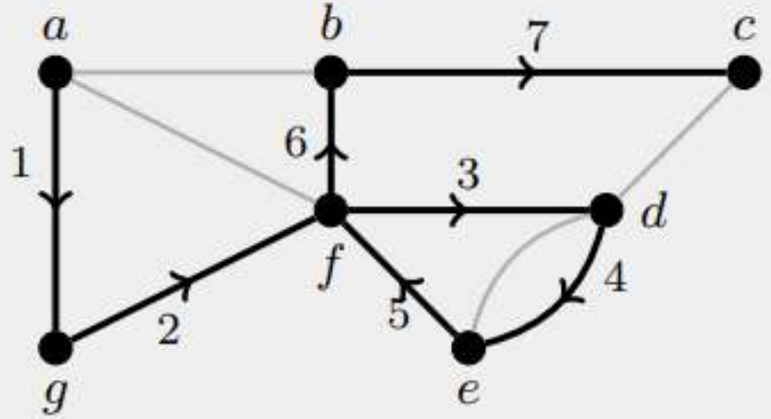
Example 2.1 Given the graph below, find a trail (that is not a path) from a to c, a path from a to c, a circuit (that is not a cycle) starting at b, and a cycle starting at b.



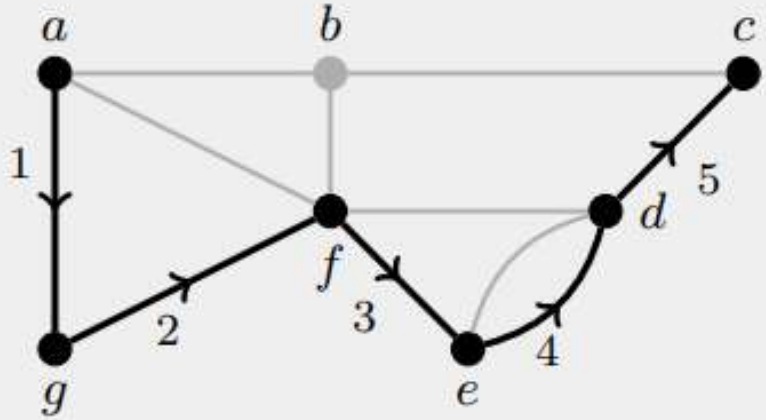

Trail from a to c
a
b
c
d
e
f
g
1
2
3
4
5
6
7

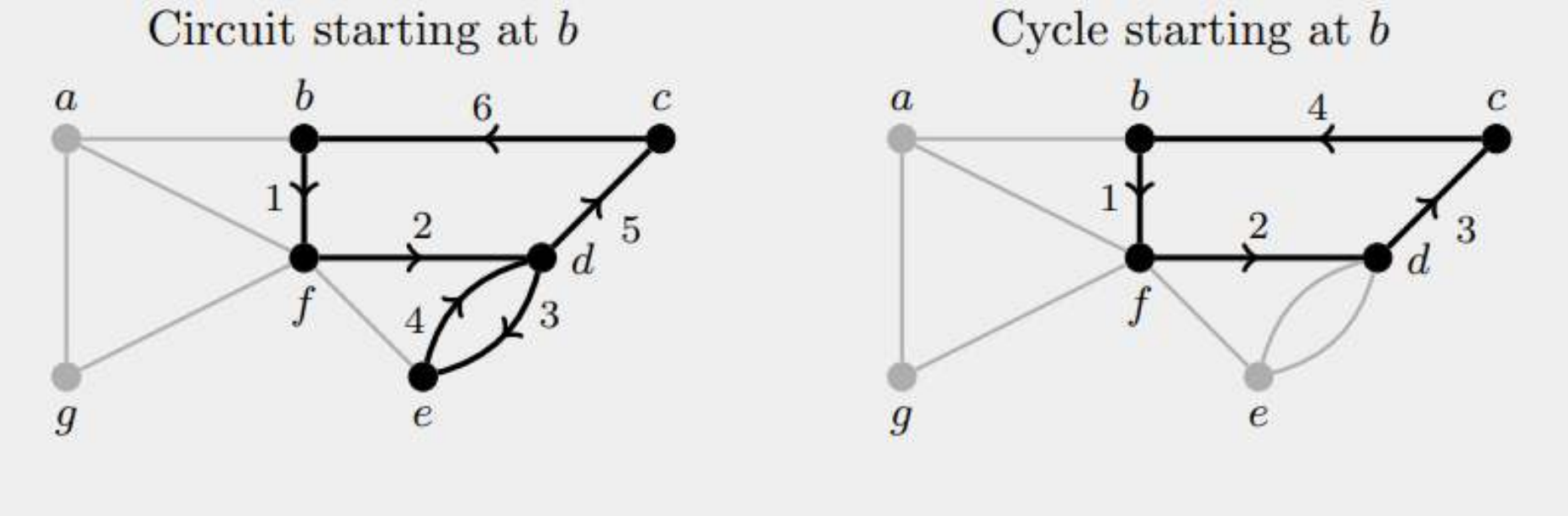

Path from a to c
a
b
c
1
2
f
3
d
4
5
g
e

Trail from a to c



Path from a to c

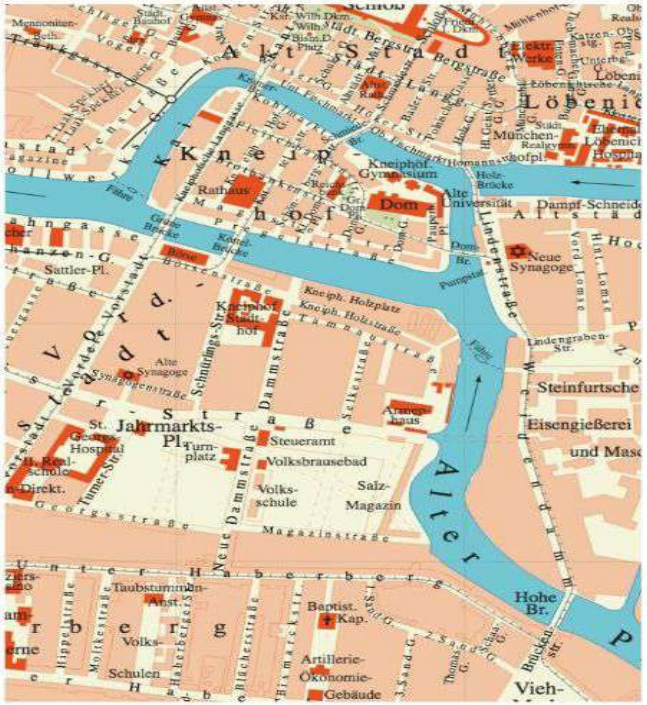



Circuit starting at b
a
b
c
d
e
f
g
1
2
3
4
5
6
Cycle starting at b
a
b
c
d
e
f
g
1
2
3
4

Definition 2.2 Let G be a graph. Two vertices x and y are ***connected*** if there exists a path from x to y in G. The graph G is ***connected*** if every pair of distinct vertices is connected.

Seven Bridges of Königsberg

Königsberg, Prussia, 1735



Scanned with CamScanner

Seven Bridges of Königsberg

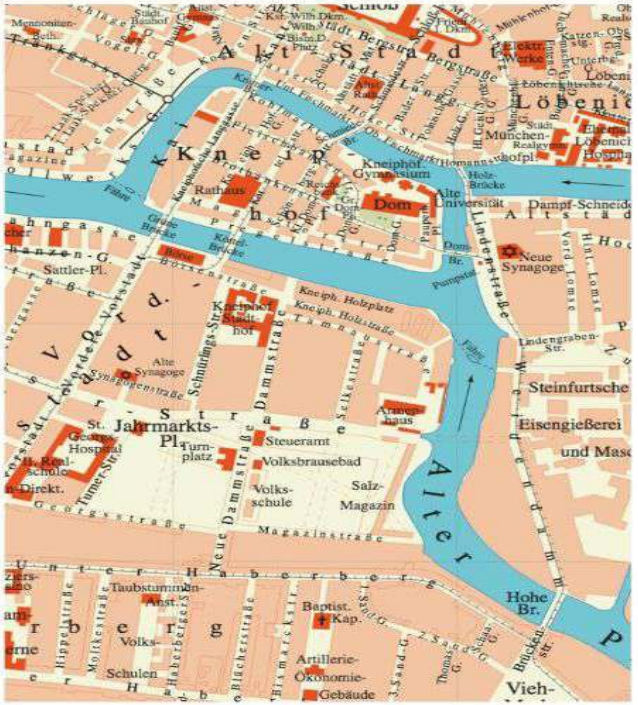
Königsberg, Prussia, 1735

Walk through Königsberg

Cross each bridge exactly once

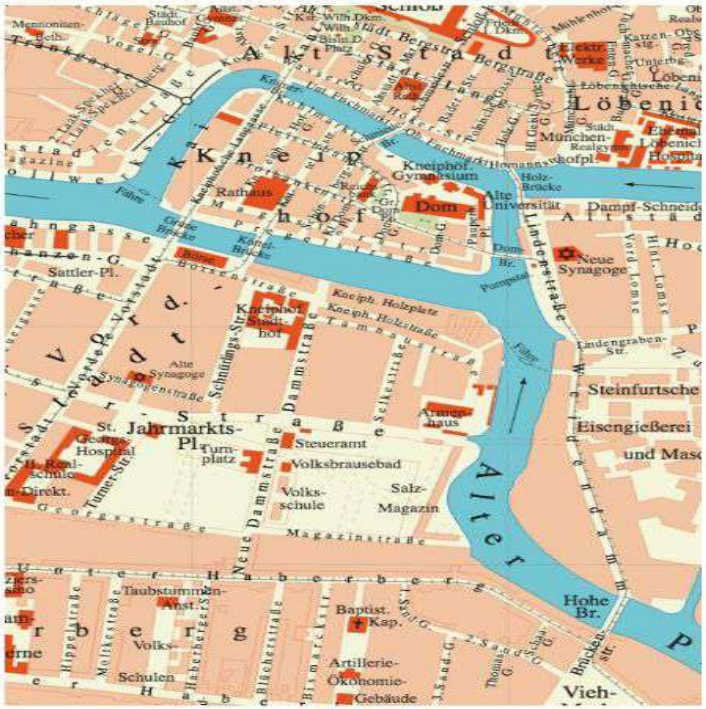
Leonhard Euler

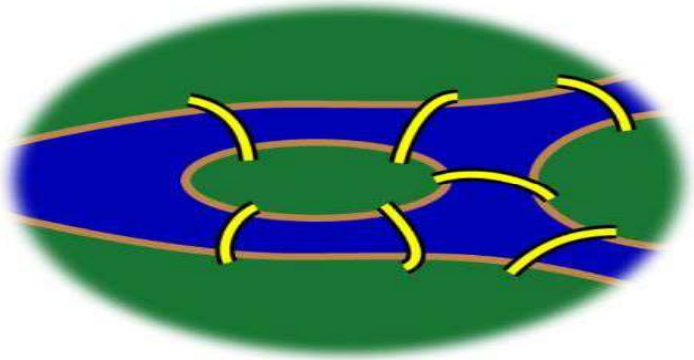
Impossible!



Scanned with CamScanner

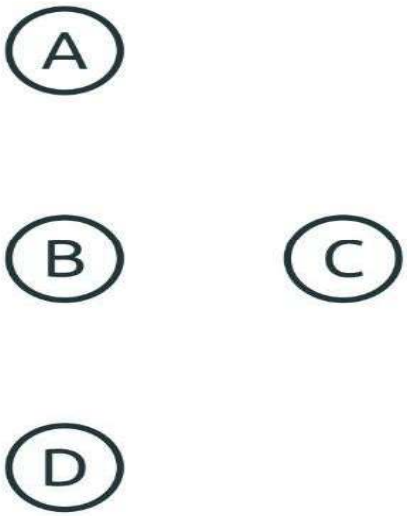
Bridges of Königsberg. Graph

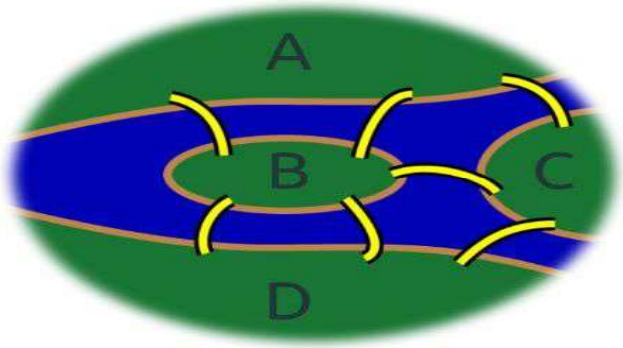




Scanned with CamScanner

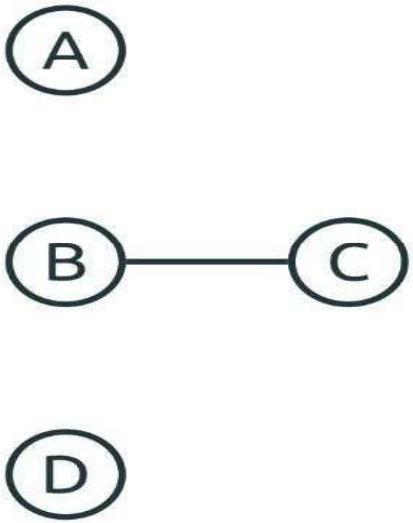
Bridges of Königsberg. Graph

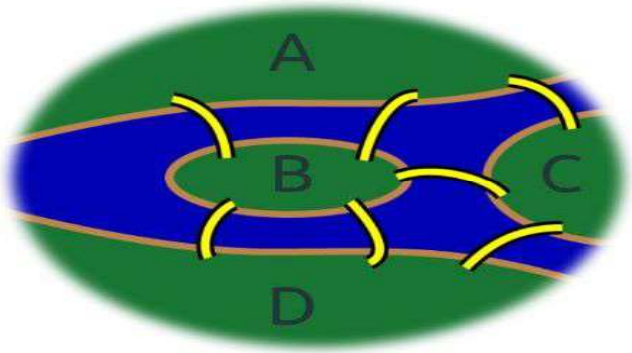




Scanned with CamScanner

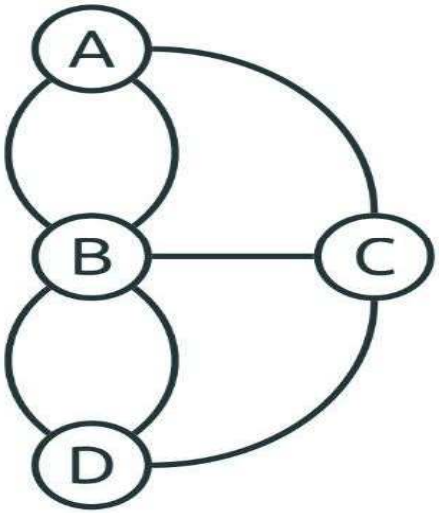
Bridges of Königsberg. Graph

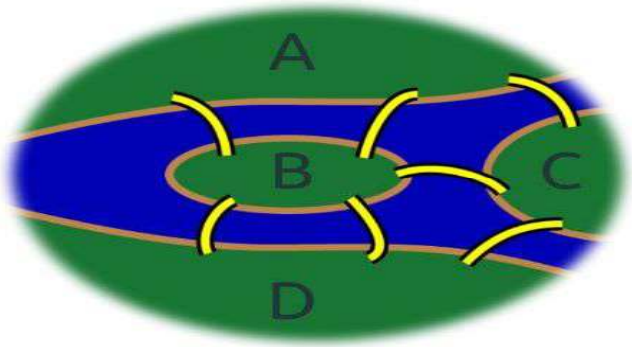




Scanned with CamScanner

Bridges of Königsberg. Graph



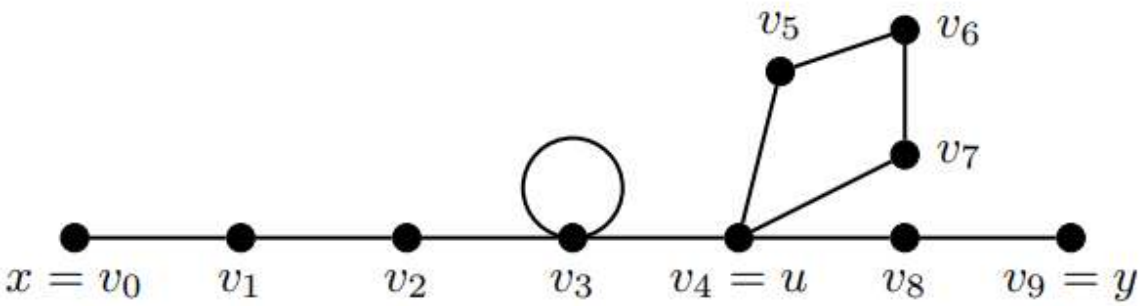


Scanned with CamScanner

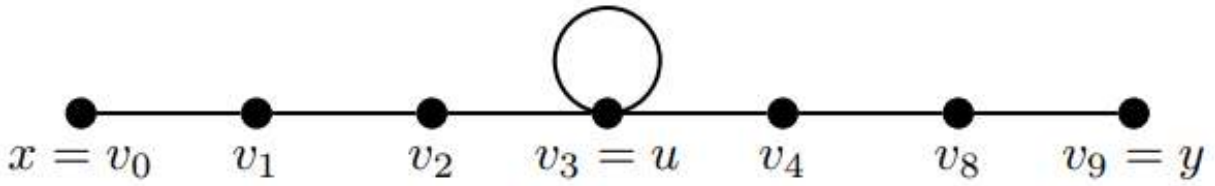
Theorem 2.3 Every $x - y$ walk contains an $x - y$ path.

Consider a walk as shown below. Using the technique from the proof above, we begin by identifying a repeated vertex, namely v_4.

$$W : v_0\ v_1\ v_2\ v_3\ v_3\ \boxed{v_4\ v_5\ v_6\ v_7\ v_4}\ v_8\ v_9$$

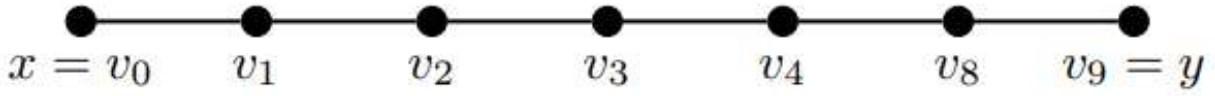


$$W' : v_0\ v_1\ v_2\ \boxed{v_3\ v_3}\ v_4\ v_8\ v_9$$



The final graph below does not contain any repeated vertices and so can be considered a path from v_0 to v_9, all of whose vertices and edges were contained in the original walk W.

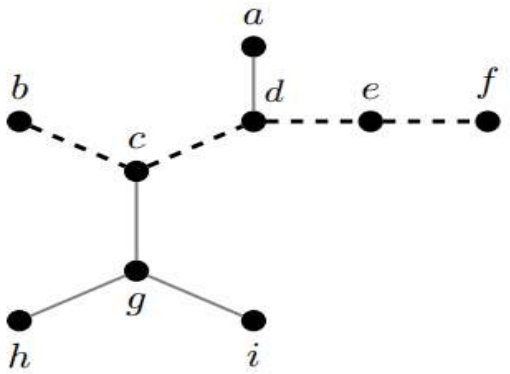
$$W'' : v_0\ v_1\ v_2\ v_3\ v_4\ v_8\ v_9$$



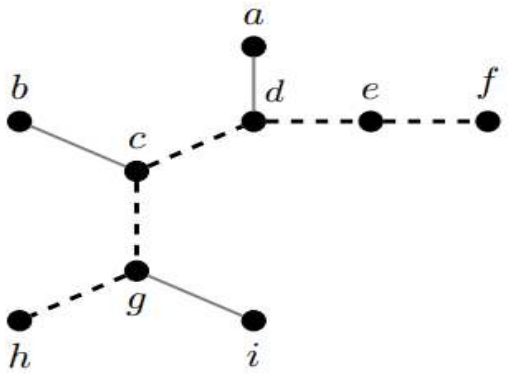
Definition 2.4 An object X is ***maximum*** if it is the largest among all objects under consideration; that is, $|X| \geq |A|$ for all $A \in \mathcal{U}$.

An object X is ***maximal*** if it cannot be made larger.

In certain scenarios, the adjectives maximum and maximal may be applied to the same object; however, this need not be true. Consider the graphs shown below. The path $b\ c\ d\ e\ f$ (highlighted on the left) is maximal because we cannot add any additional vertices and keep it a path. However, it is not maximum since a longer path can be found, namely $h\ g\ c\ d\ e\ f$ shown on the right. Note that this second path is a maximum path within the graph shown.



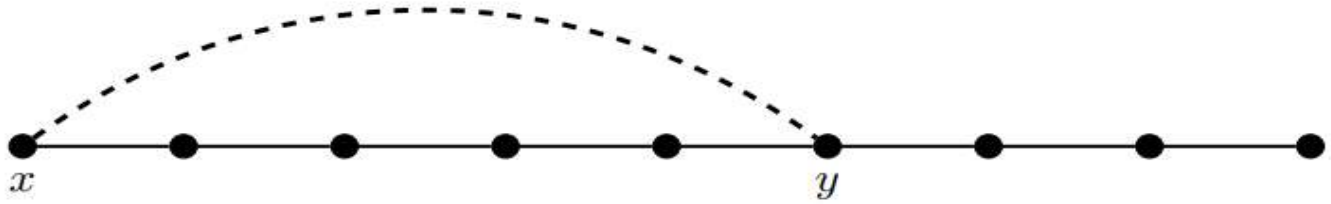
Maximal $b - f$ path



Maximum path in G

Theorem 2.5 If every vertex of a graph has degree at least 2 then G contains a cycle.

Proof: Let P be a maximal path in G and let x be an endpoint of P. Since P is maximal, it cannot be extended and so every neighbor of x must already be a vertex of P. Since x has degree at least 2, we know there must be another neighbor y of x in $V(P)$ via an edge not a part of the path.



Then the edge xy completes a cycle with the portion of P from x to y.

Definition 2.6 Let G be a graph. An ***eulerian circuit*** (or ***trail***) is a circuit (or trail) that contains every edge and every vertex of G.

If G contains an eulerian circuit it is called ***eulerian*** and if G contains an eulerian trail but not an eulerian circuit it is called ***semi-eulerian***.

Theorem 2.7 A graph G is eulerian if and only if

(i) G is connected and

(ii) every vertex has even degree.

Corollary 2.8 A graph G is semi-eulerian if and only if

(i) G is connected and

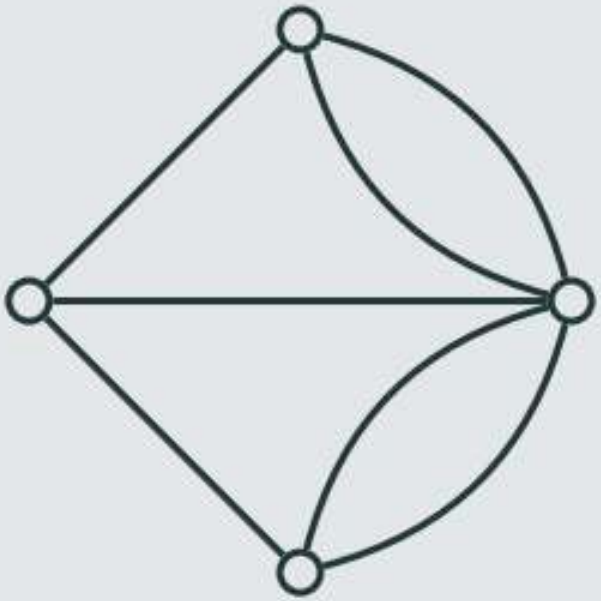
(ii) exactly two vertices have odd degree.

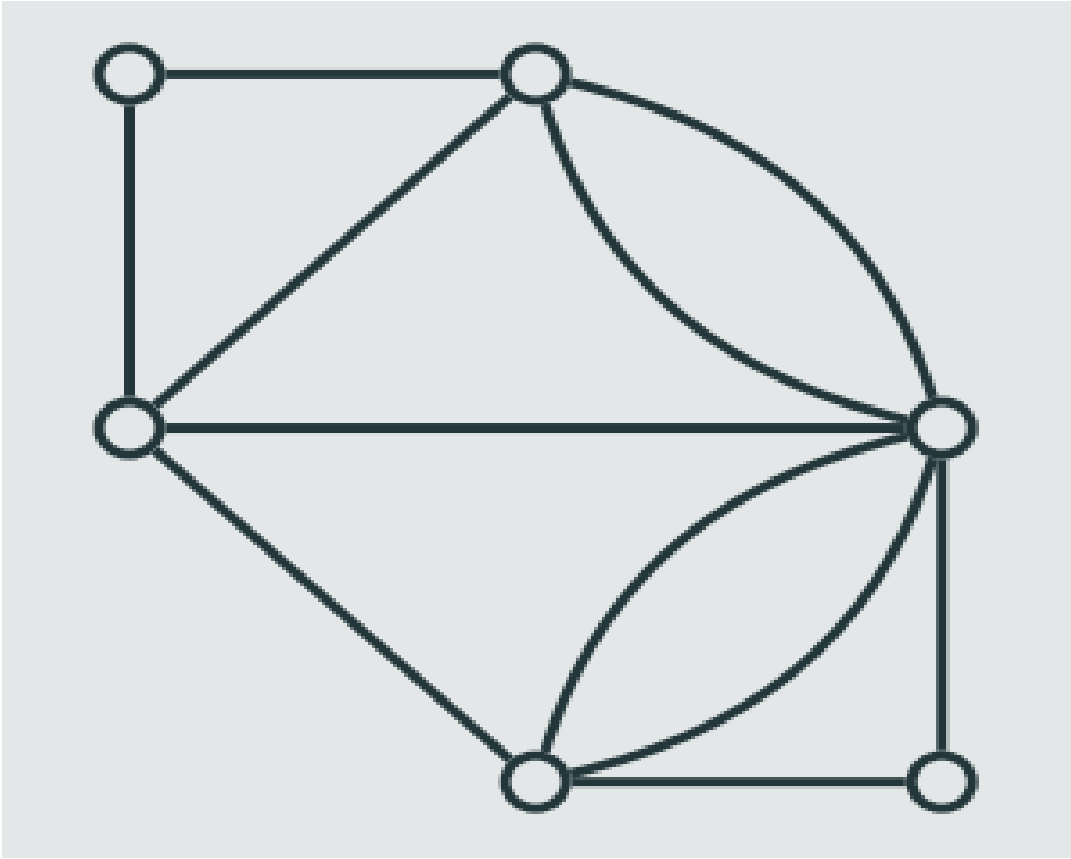
Criteria

Theorem

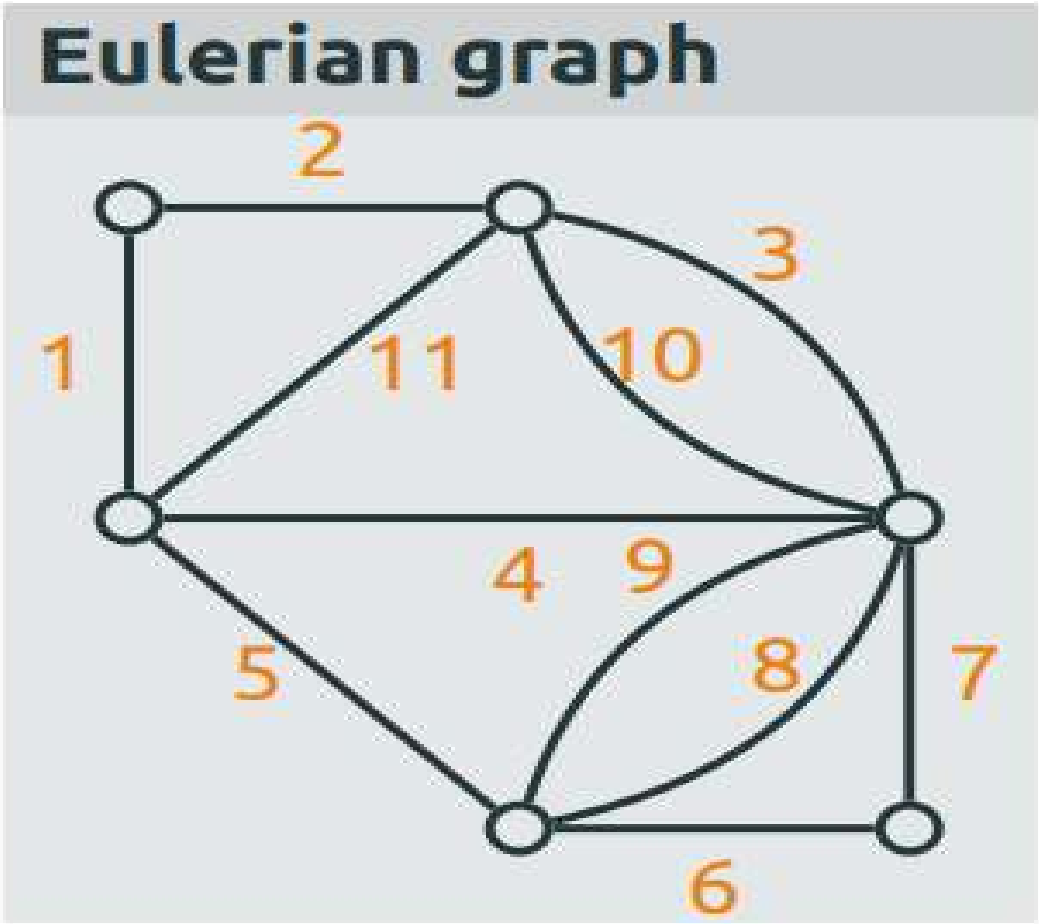
A connected *undirected* graph contains an Eulerian cycle, if and only if the degree of every node is even.

Non-Eulerian graph





Eulerian graph



Criteria

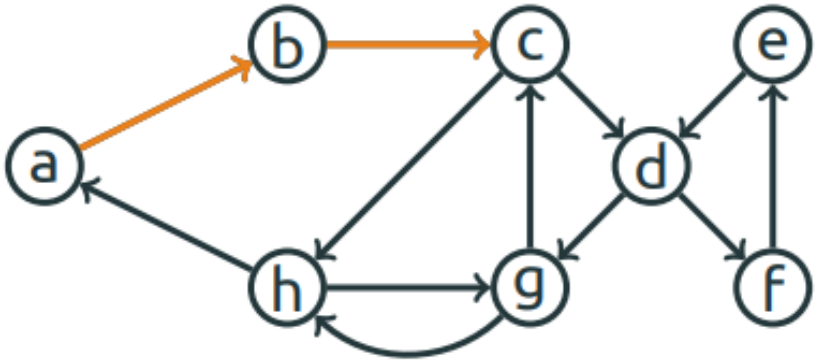
Theorem

A connected *undirected* graph contains an Eulerian cycle, if and only if the degree of every node is even.

Theorem

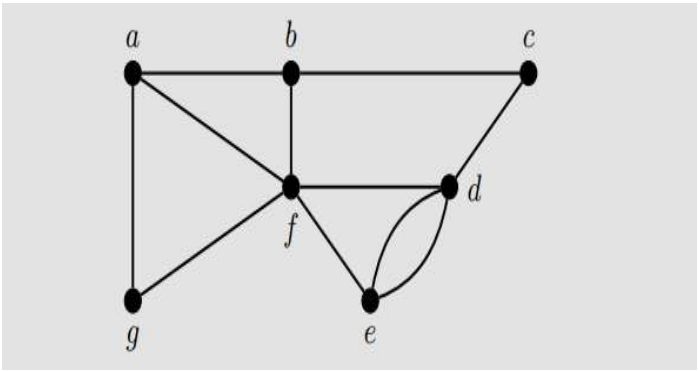
A strongly connected *directed* graph contains an Eulerian cycle, if and only if, for every node, its in-degree is equal to its out-degree.

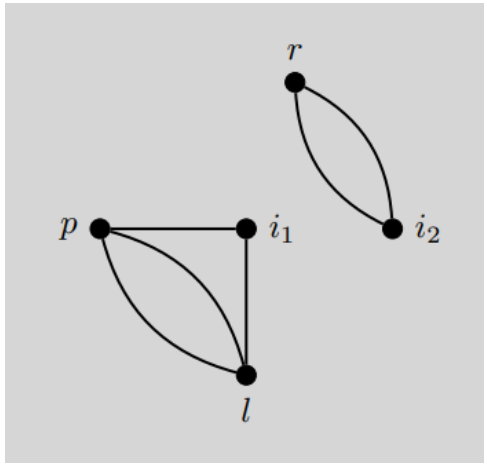
Proof (Directed Case)

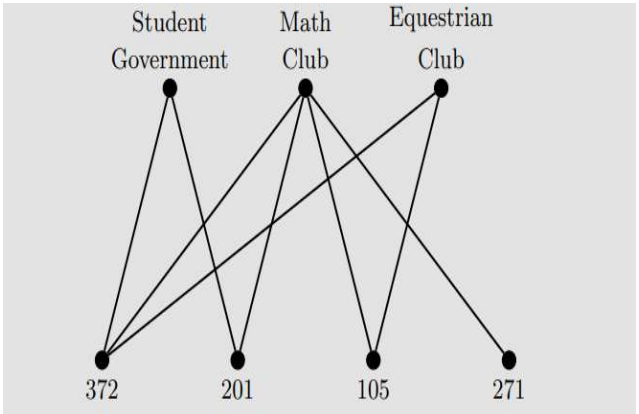


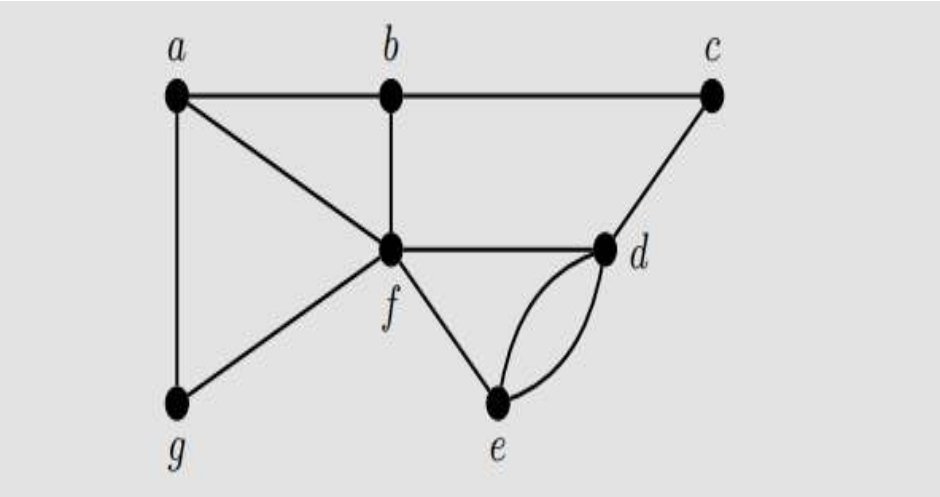
since the graph is balanced, at some point we'll return back to the starting node

Example 2.3 Consider the graphs appearing in the examples from this and the previous chapter. Which ones are eulerian? semi-eulerian? neither?

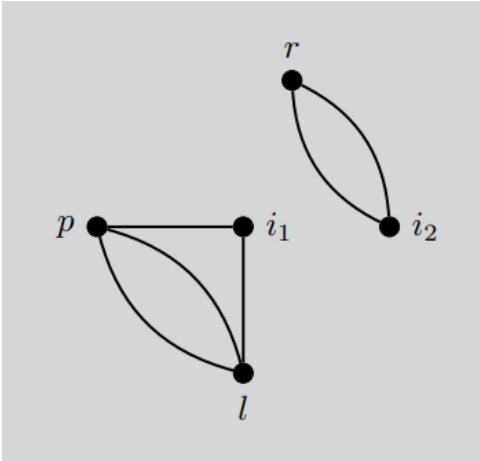




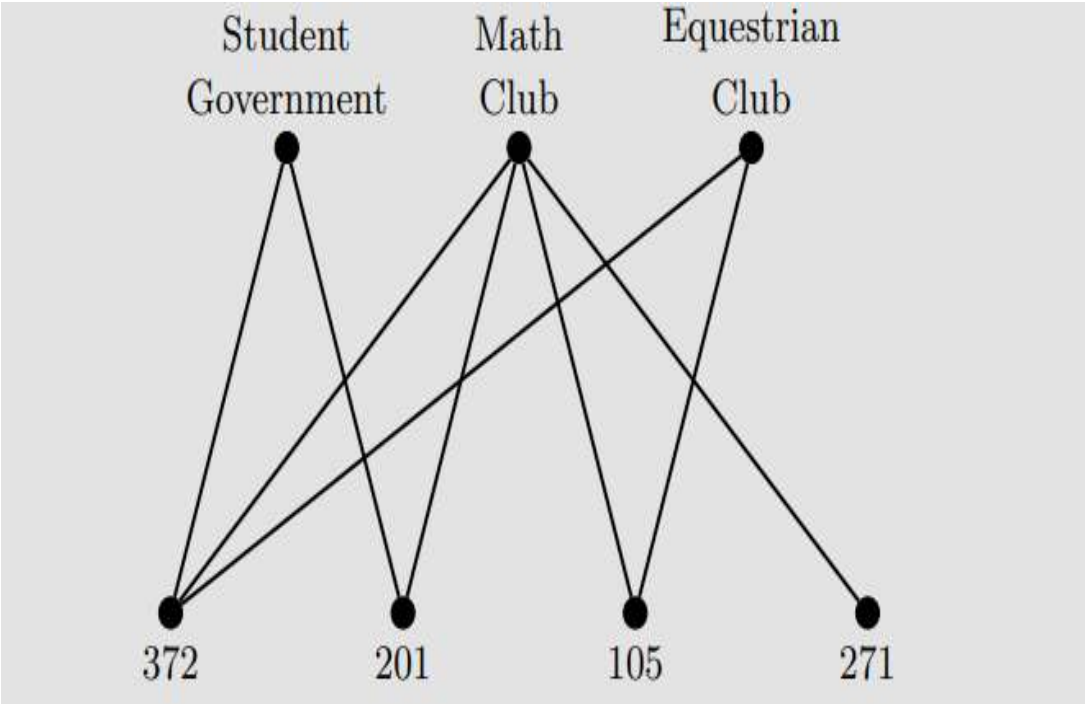




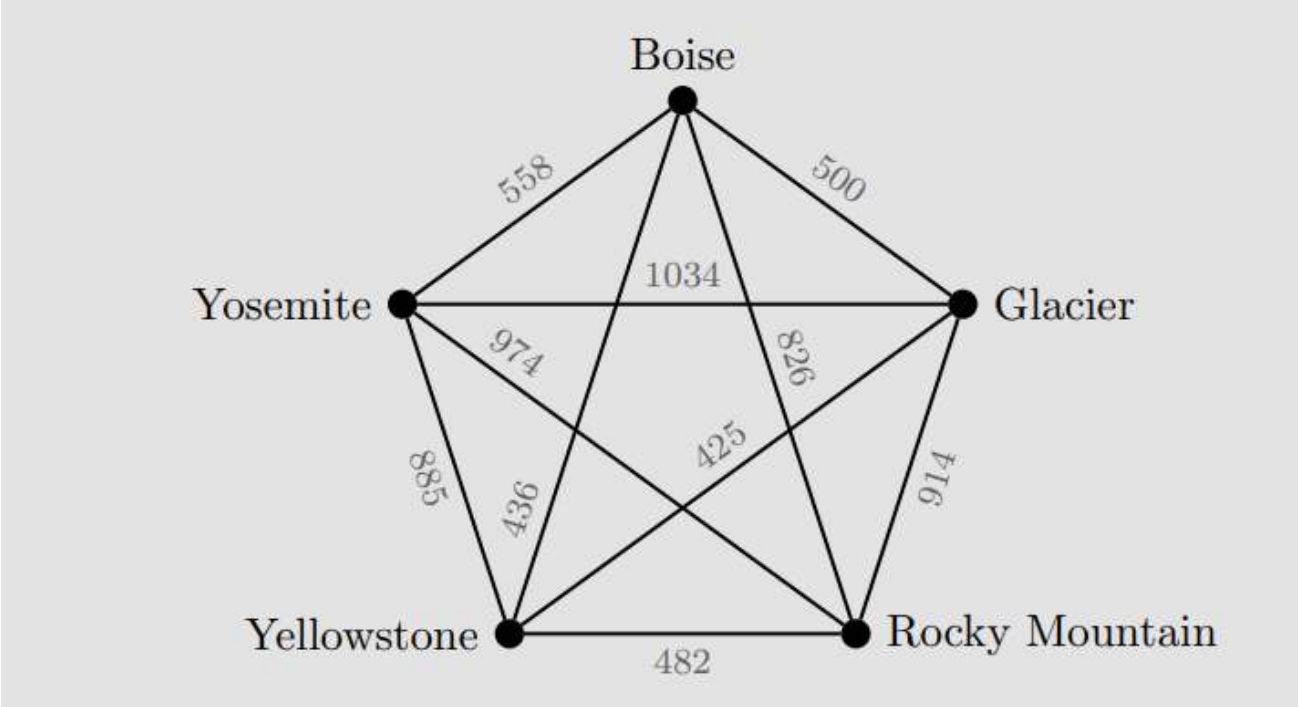
Even though the graph in Example 2.1 is connected, it is neither eulerian nor semi-eulerian since it has more than two odd vertices (namely, $a, b, e,$ and f).

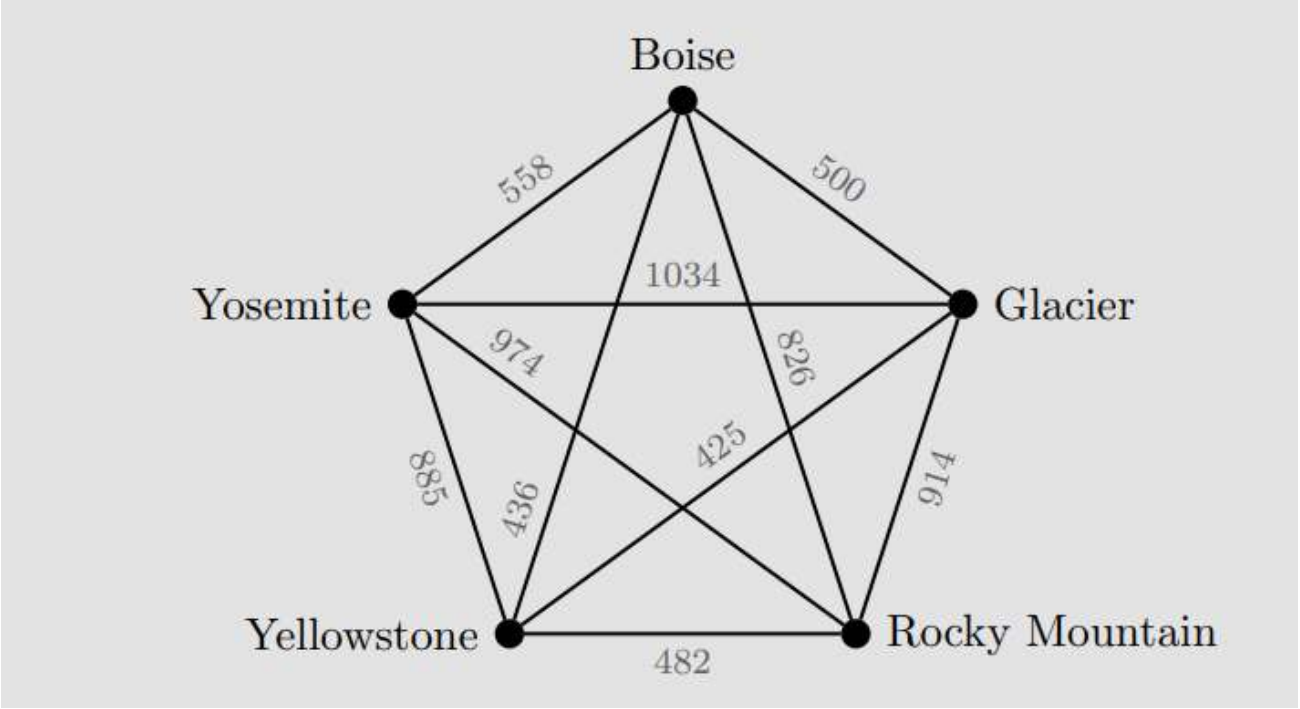


- The graph representing Island City in Example 2.2 is not connected, so it is neither eulerian nor semi-eulerian.



- The graph in Example 1.11 is semi-eulerian since it is connected and exactly two vertices are odd (namely, 372 and 271).


Boise
Yosemite
Glacier
Yellowstone
Rocky Mountain
558
500
1034
974
826
885
436
425
914
482



- The graph in Example 1.7 is eulerian since it is connected and all the vertices have degree 4.

National University of computer and emerging sciences

Algorithms of Finding Eulerian Circuit (or trails)

Foundation for Advancement of Science and Technology

Fleury's Algorithm

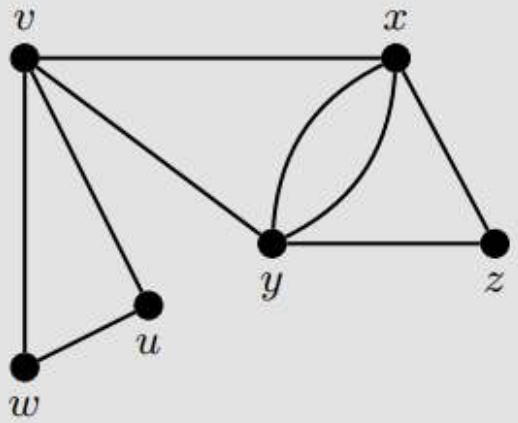
Input: Connected graph G where zero or two vertices are odd.

Steps:

1. Choose a starting vertex, call it v. If G has no odd vertices, then any vertex can be the starting point. If G has exactly two odd vertices, then v must be one of the odd vertices.
2. Choose an edge incident to v that is unlabeled and label it with the number in which it was chosen, ensuring that the graph consisting of unlabeled edges remains connected.
3. Travel along the edge to its other endpoint.
4. Repeat Steps (2) and (3) until all edges have been labeled.

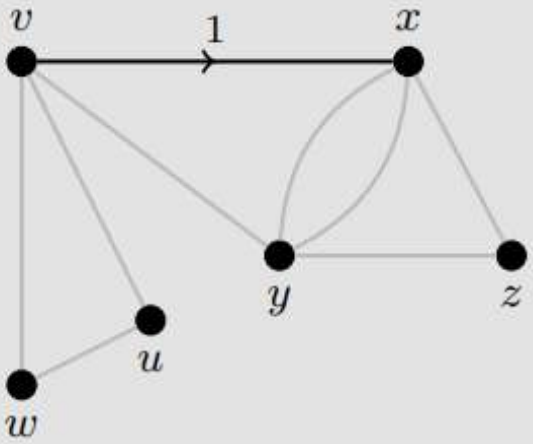
Output: Labeled eulerian circuit or trail.

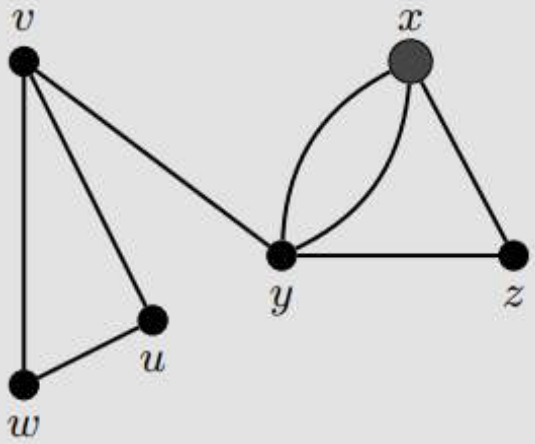
Example 2.4 Input: A connected graph (shown below) where every vertex has even degree. We are looking for an eulerian circuit.



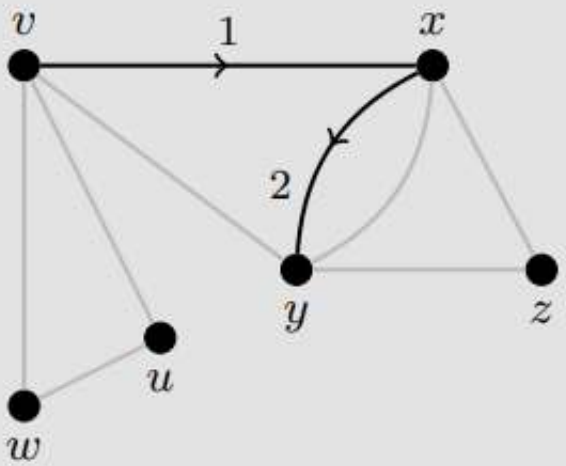
Step 1: Since no starting vertex is explicitly stated, we choose vertex v to be the starting vertex.

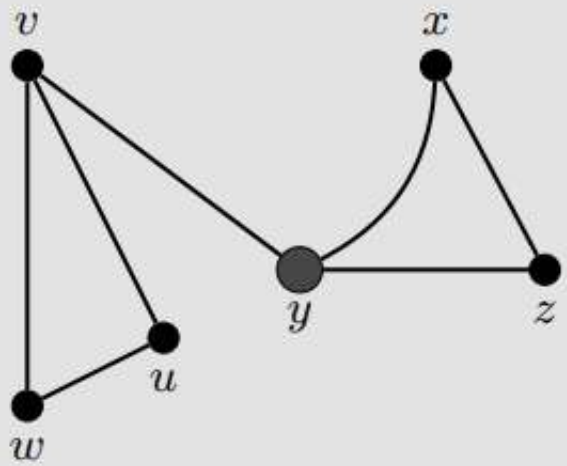
Step 2: We can choose any edge incident to v. Here we chose vx. The labeled graph is on the left and the unlabeled portions are shown on the right with edges removed that have already been chosen.



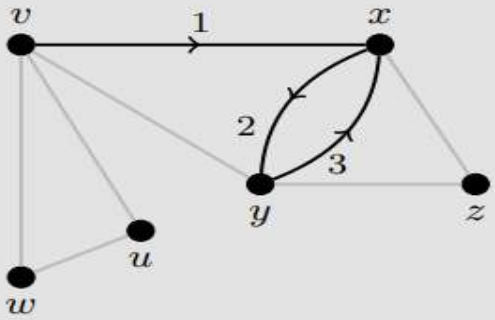


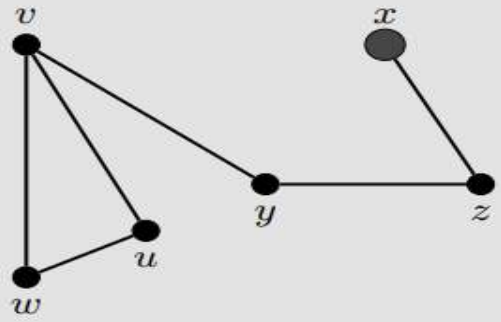
Step 3: Looking at the graph to the right, we can choose any edge out of x. Here we chose xy. The labeled and unlabeled graphs have been updated below.



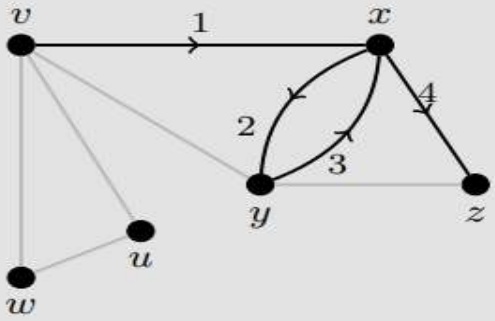


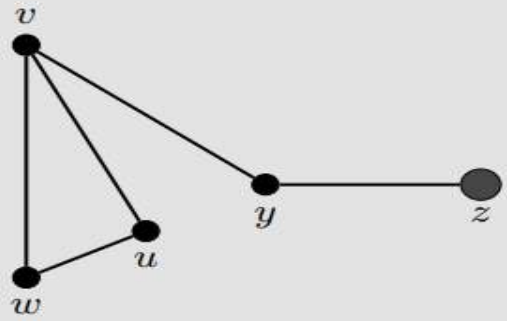
Step 4: At this point we cannot choose yv, as its removal would disconnect the unlabeled graph shown on the right in Step 3. However, yx and yz are both valid choices. Here we chose yx.



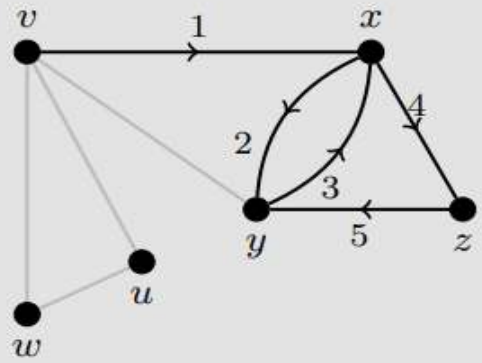


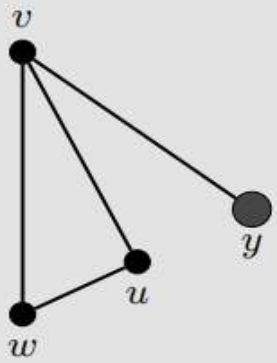
Step 5: There is only one available edge xz.



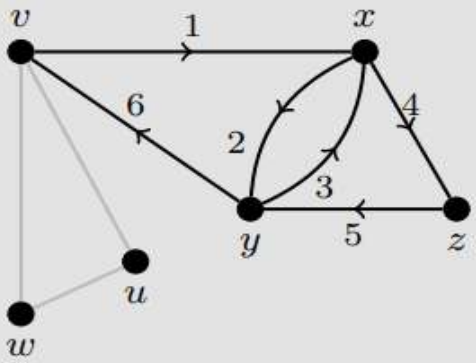


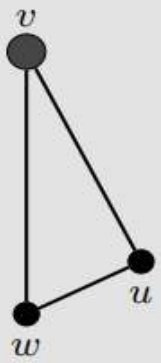
Step 6: There is only one available edge zy.



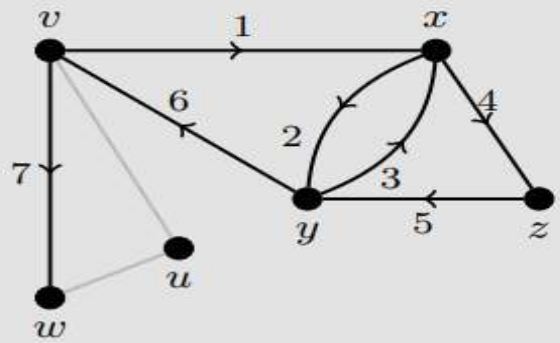


Step 7: There is only one available edge yv.

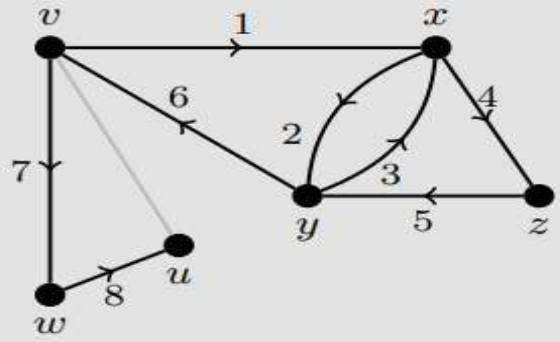




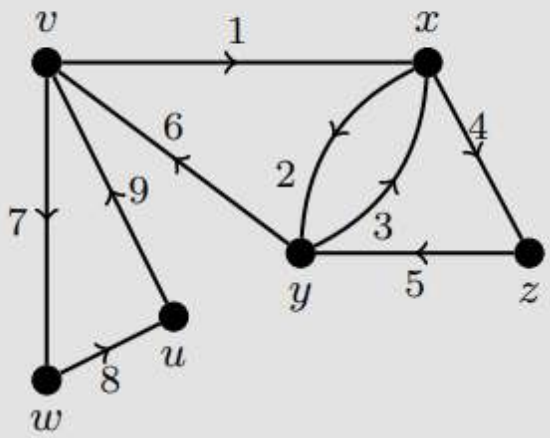
Step 8: Both vw and vu are valid choices for the next edge. Here we chose vw.



Step 9: There is only one available edge wu.



Step 10: There is only one available edge uv.



Output: The graph above on the left has an eulerian circuit labeled, starting and ending at vertex v.

Hierholzer's Algorithm

Input: Connected graph G where all vertices are even.

Steps:

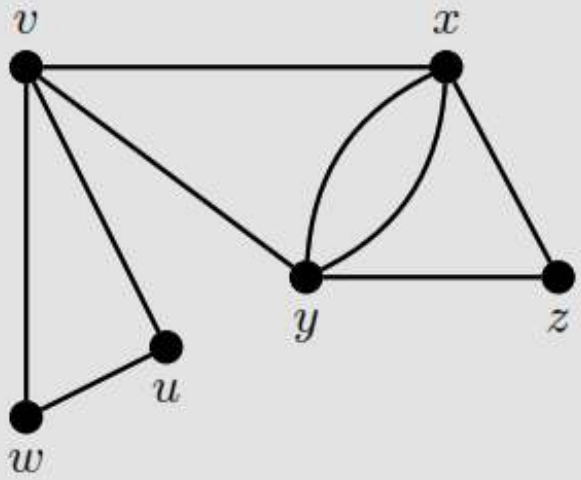
1. Choose a starting vertex, call it v. Find a circuit C originating at v.
2. If any vertex x on C has edges not appearing in C, find a circuit C' originating at x that uses two of these edges.
3. Combine C and C' into a single circuit C^*.
4. Repeat Steps (2) and (3) until all edges of G are used.

Output: Labeled eulerian circuit.

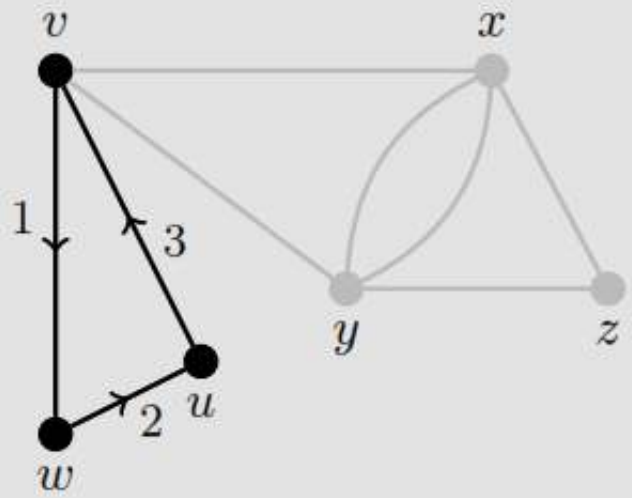
NOTE:

Hierholzer's Algorithm requires the graph to be **eulerian**,
whereas ***Fleury's Algorithm*** allows for the graph to be eulerian or semi-eulerian.

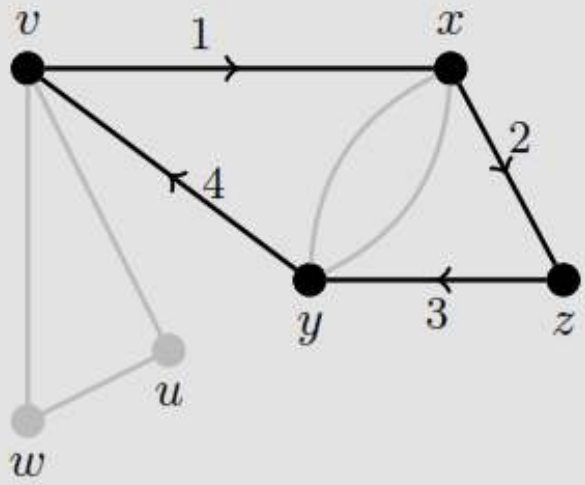
Example 2.5 Input: A connected graph where every vertex has even degree. We are looking for an eulerian circuit.



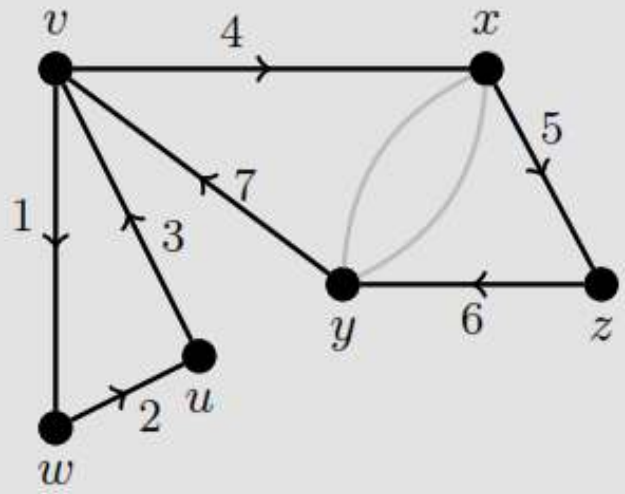
Step 1: Since no starting vertex is explicitly stated, we choose v and find a circuit originating at v. One such option is highlighted below.



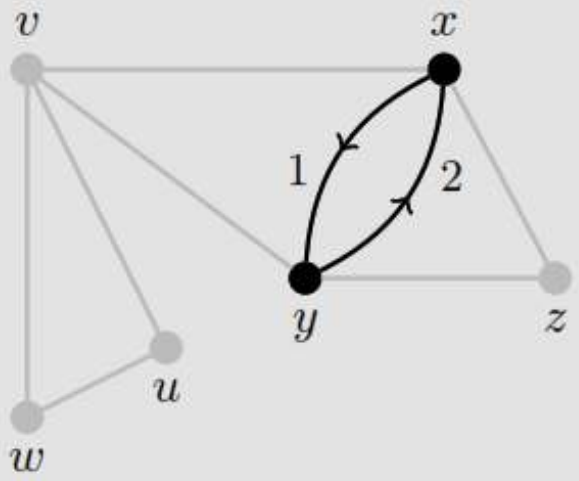
Step 2: As $\deg(v) = 4$ and two edges remain for v (shown in gray in the previous figure), a second circuit starting at v is needed. One option is shown below.



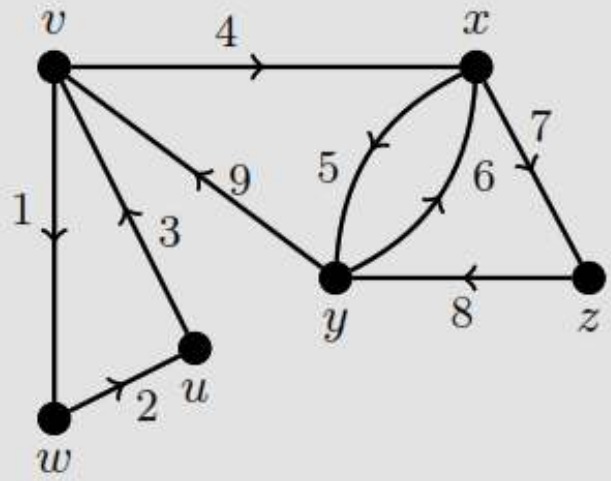
Step 3: Combine the two circuits from Step 1 and Step 2. There are multiple ways to combine two circuits, but it is customary to travel the first circuit created and then travel the second.



Step 4: As $\deg(x) = 4$ and two edges remain for x (shown in gray above), a circuit starting at x is needed. It is shown below.



Step 5: Combine the two circuits from Step 3 and Step 4.



Output: The graph above gives a labeled eulerian circuit originating at v.

- ***There are advantages and disadvantages for these two methods; in particular, both Fleury's and Hierholzer's Algorithms will find an eulerian circuit when one exists, whereas only Fleury's can be used to find an eulerian trail (see Exercise 2.10 for a modification of Hierholzer's that will find an eulerian trail)***

- ***Try a few more examples and make additional comparisons. In practice, either algorithm is a good choice for finding an eulerian circuit. Pick the method that works best for you.***

Go through section 2.1.5

Definition 2.10 A cycle in a graph G that contains every vertex of G is called a ***hamiltonian cycle***. A path that contains every vertex is called a ***hamiltonian path***. A graph that contains a hamiltonian cycle is called ***hamiltonian***.

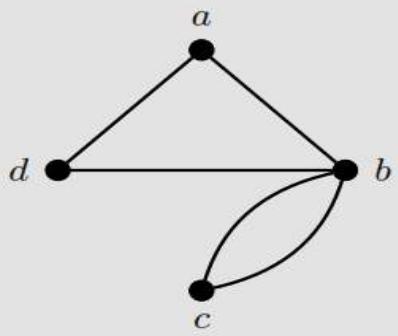
Hamiltonian Cycle

A Hamiltonian cycle visits every node of a graph exactly once.

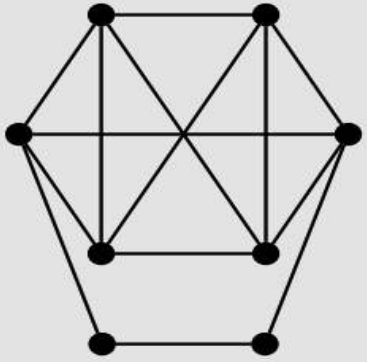
Simple Criteria?

- No simple criteria is known for the Hamiltonian cycle problem

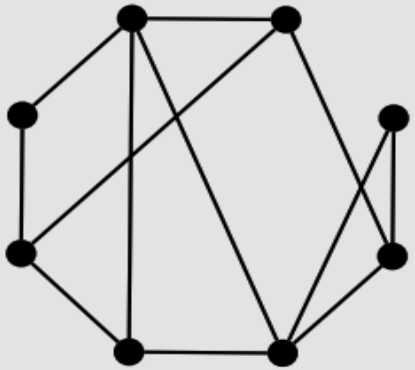
Example 2.8 For each of the graphs below, determine if they have hamiltonian cycles (and paths) and eulerian circuits (and trails).



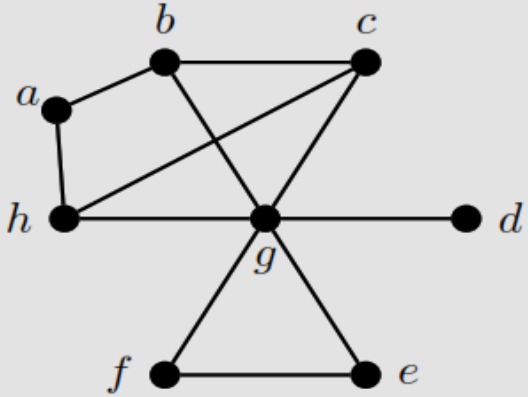
G_1



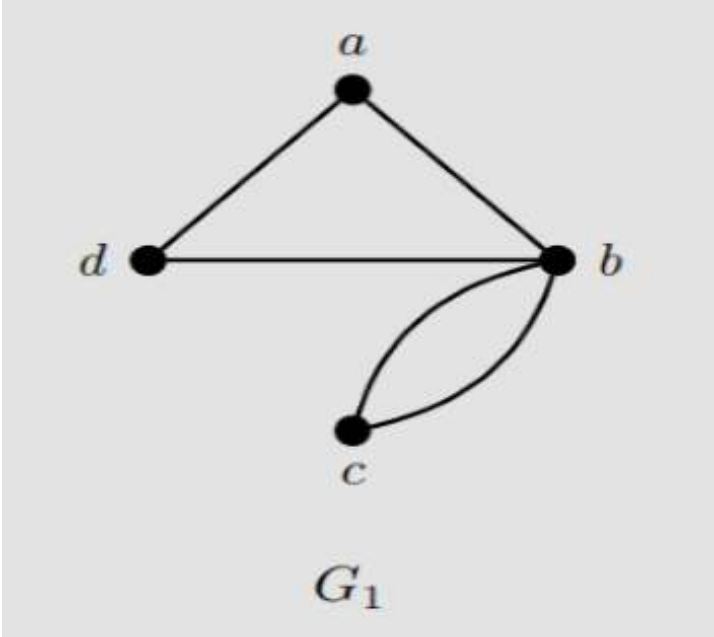
G_2



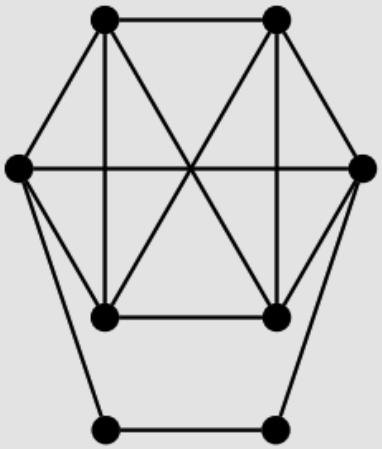
G_3



G_4

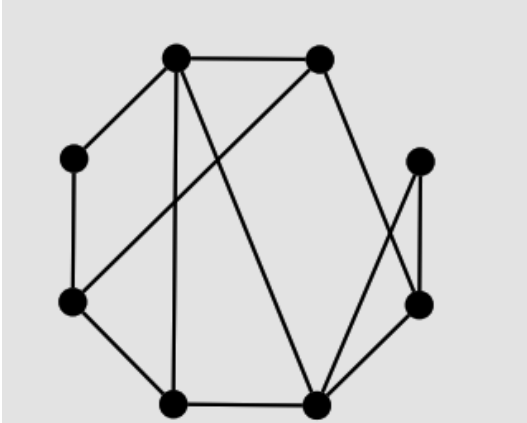


Solution: Since G_1 is connected and all vertices are even, we know it has an eulerian circuit. There is no hamiltonian cycle since we need to include c in the cycle and by doing so we have already passed through b twice, making it impossible to visit a and d.



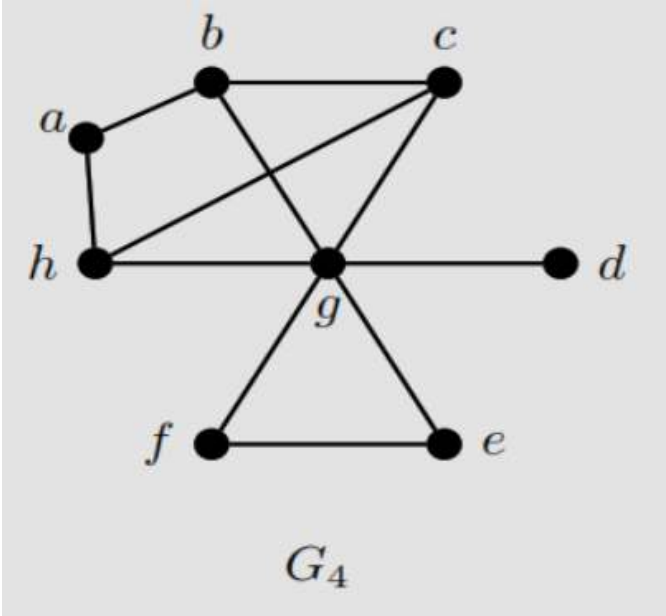
G_2

Since G_2 is connected and all vertices are even, we know it is eulerian. Hamiltonian cycles and hamiltonian paths also exist. To find one such path, remove any one of edges from a hamiltonian cycle.



G_3

Some vertices of G_3 are odd, so we know it is not eulerian. Moreover, since more than two vertices are odd, the graph is not semi-eulerian. However, this graph does have a hamiltonian cycle (and so also a hamiltonian path). Can you find it?



G_4

Four vertices of G_4 are odd, we know it is neither eulerian nor semi-eulerian. This graph does not have a hamiltonian cycle since d cannot be a part of any cycle. Moreover, this graph does not have a hamiltonian path since any traversal of every vertex would need to travel through g multiple times.

- If a graph has a hamiltonian cycle, it automatically has a hamiltonian path (just leave off the last edge of the cycle to obtain a path).
- If a graph has a hamiltonian path, it may or may not have a hamiltonian cycle.

Properties of Hamiltonian Graphs

(1) G must be connected.

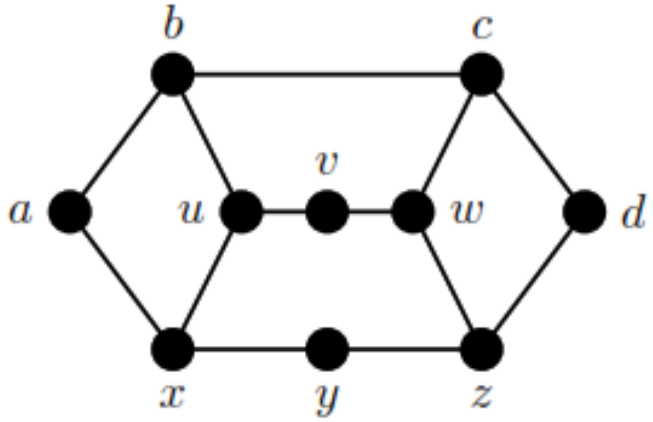
(2) No vertex of G can have degree less than 2.

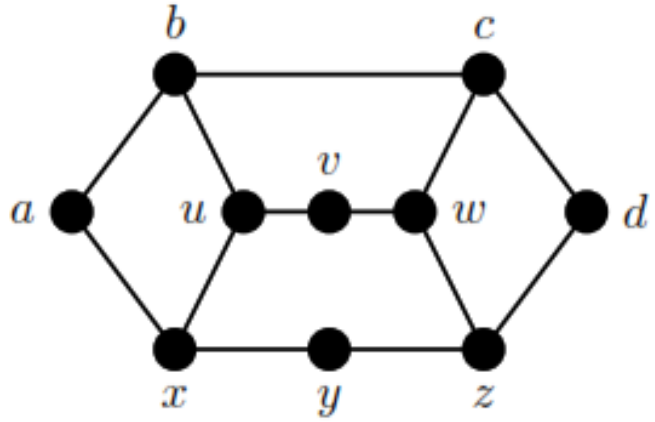
(3) G cannot contain a ***cut-vertex***, that is a vertex whose removal disconnects the graph.

(4) If G contains a vertex x of degree 2 then both edges incident to x must be included in the cycle.

(5) If two edges incident to a vertex x must be included in the cycle, then all other edges incident to x cannot be used in the cycle.

(6) If in the process of attempting to build a hamiltonian cycle, a cycle is formed that does not span G, then G cannot be hamiltonian.


b
c
v
a
u
w
d
x
y
z

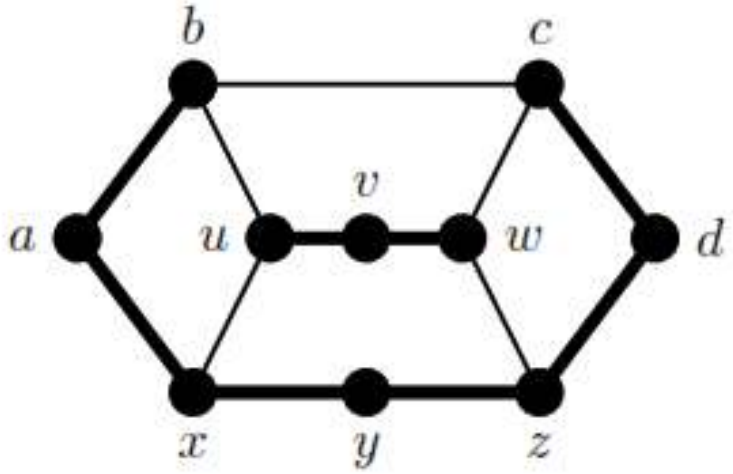


There are four vertices of degree two in G, namely, a, d, v and y. If G has a Hamiltonian cycle C, then by rule (i), the eight edges $ab, ax, cd, dz, uv, vw, xy$ and yz are required edges in $E(C)$ as indicated in bold in Fig. 5.1.4(b).

It is now obvious that the set of eight required edges can be expanded to form such a C by adding bu and cw. Thus G is a Hamiltonian graph.

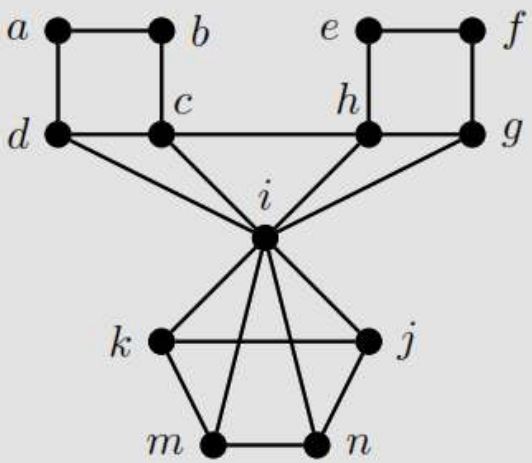
There are four vertices of degree two in G, namely, a, d, v and y. If G has a Hamiltonian cycle C, then by rule (i), the eight edges $ab, ax, cd, dz, uv, vw, xy$ and yz are required edges in $E(C)$ as indicated in bold in Fig. 5.1.4(b).

It is now obvious that the set of eight required edges can be expanded to form such a C by adding bu and cw. Thus G is a Hamiltonian graph.

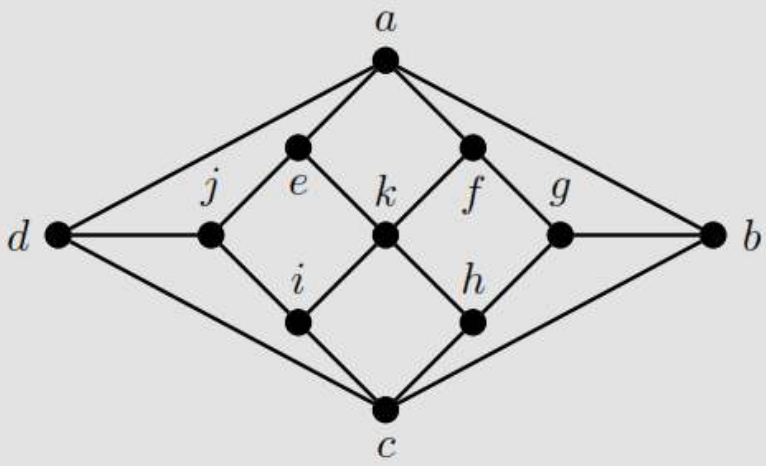


(b)

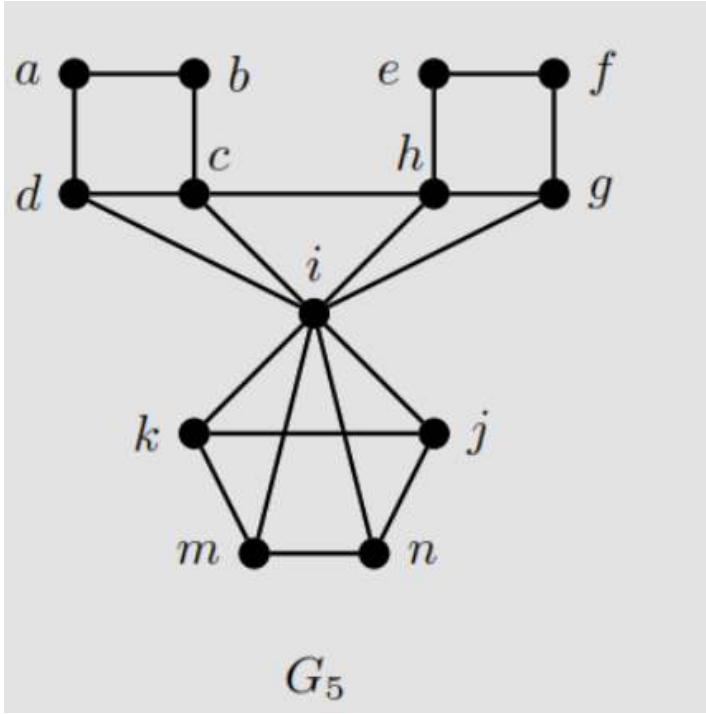
Example 2.9 Use the properties listed above to show that the graphs below are not hamiltonian.



G_5

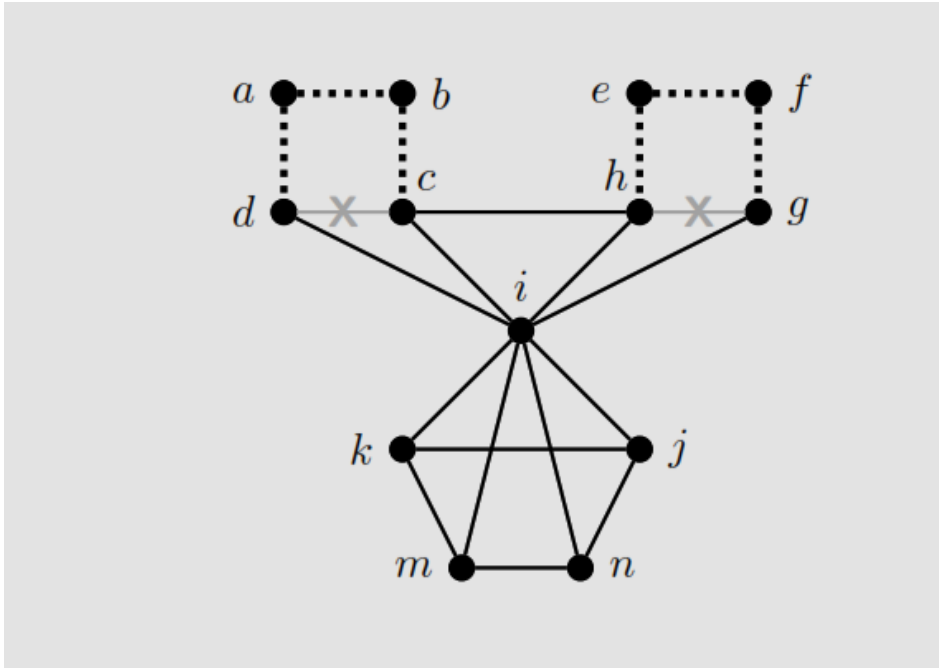


G_6

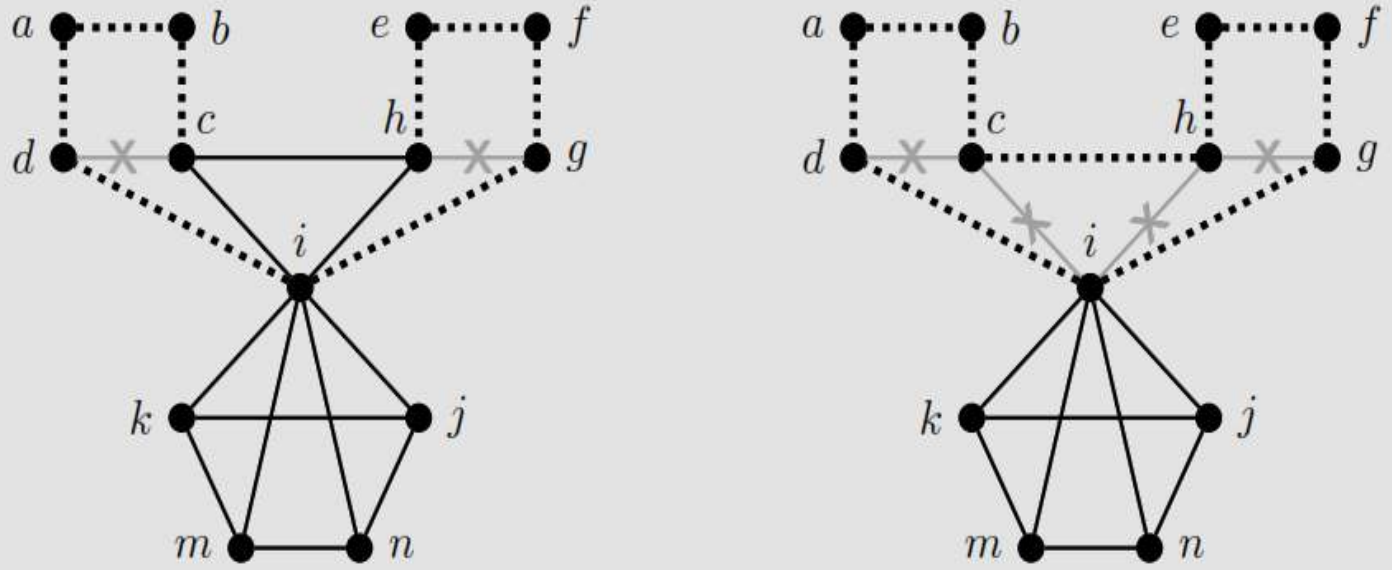


G_5

Solution: For G_5, notice that vertices a, b, e, and f all have degree 2 and so all the edges incident to these vertices must be included in the cycle. But then edges cd and hg cannot be a part of a cycle since they would create smaller cycles that do not include all of the vertices in G_5.

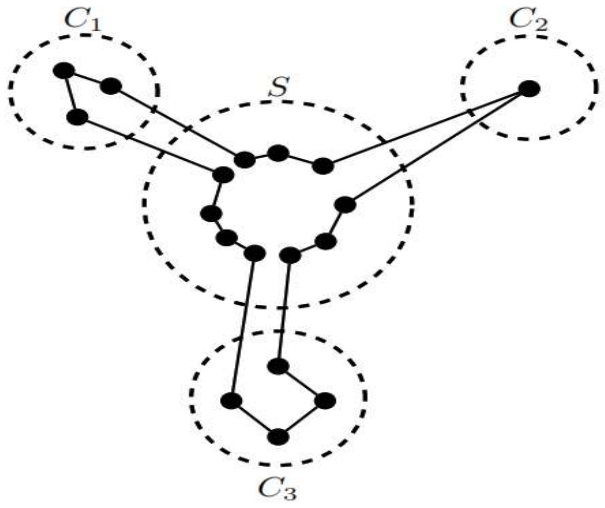

a
b
e
f
c
h
d
g
i
k
j
m
n

Since only one edge remains incident to d and g, namely di and gi, then these must be a part of the cycle. But in doing so, we would be forced to use ch since the other edges incident to i could not be chosen. This creates a cycle that does not span G_5, and so G_5 is not hamiltonian.



Proposition 2.11 If G is a graph with a hamiltonian cycle, then $G - S$ has at most $|S|$ components for every nonempty set $S \subseteq V$.

Proof: Assume G has a hamiltonian cycle C and S is a nonempty subset of vertices. Then as we traverse C we will leave and return to S from distinct vertices in S since no vertex can be used more than once in a cycle. Since C must span $V(G)$, we know S must have at least as many vertices as the number of components of $G - S$.

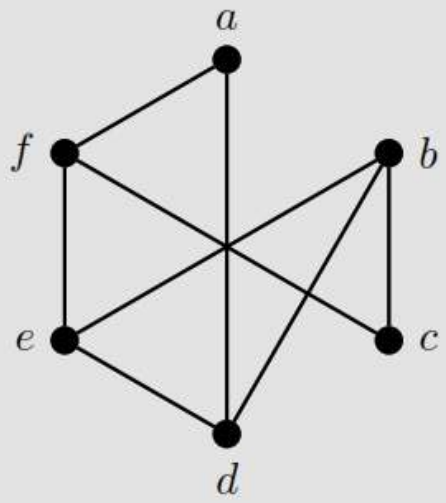


Theorem 2.12 (Dirac's Theorem) Let G be a graph with $n \geq 3$ vertices. If every vertex of G satisfies $\deg(v) \geq \frac{n}{2}$, then G has a hamiltonian cycle.

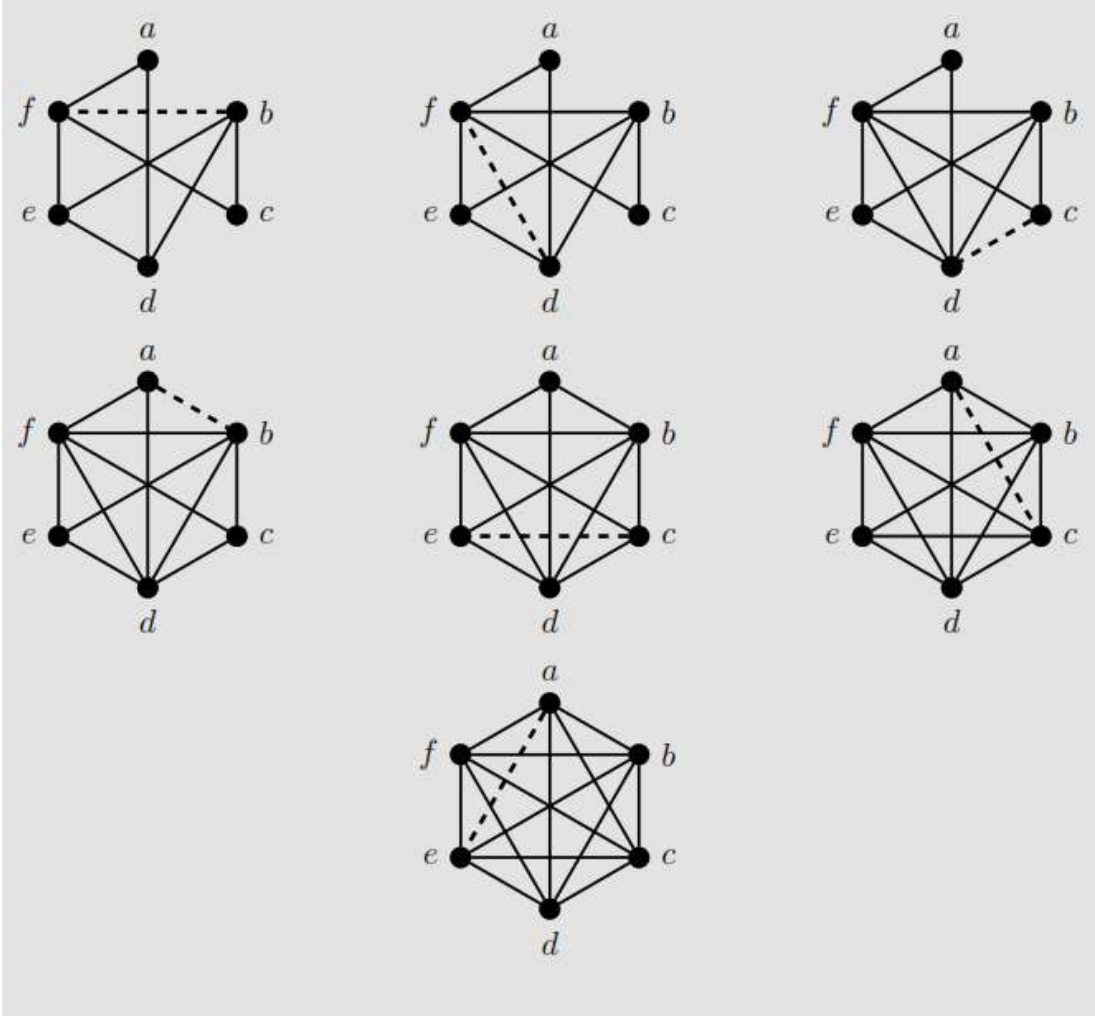
Theorem 2.13 (Ore's Theorem) Let G be a graph with $n \geq 3$ vertices. If $\deg(x) + \deg(y) \geq n$ for all distinct nonadjacent vertices, then G has a hamiltonian cycle.

Definition 2.14 The ***hamiltonian closure*** of a graph G, denoted $cl(G)$, is the graph obtained from G by iteratively adding edges between pairs of nonadjacent vertices whose degree sum is at least n, that is we add an edge between x and y if $\deg(x) + \deg(y) \geq n$, until no such pair remains.

Example 2.10 Find a hamiltonian closure for the graph below.



Solution: First note that the degrees of the vertices are $2, 3, 2, 3, 3, 3$ (in alphabetic order). Since the closure of G is formed by putting an edge between nonadjacent vertices whose degree sum is at least 6, we must begin with edge bf or df. We chose to start with bf. In the sequence of graphs shown on the next page, the edge added at each stage is a thick dashed line.


a
f
b
e
c
d