Chapter 11

Graph Theory

"The origins of graph theory are humble, even frivolous." (N. Biggs, E. K. Lloyd, and R. J. Wilson)

Let us start with a formal definition of what is a graph.

Definition 72. A graph $G = (V, E)$ is a structure consisting of a set V of vertices (also called nodes), and a set E of edges, which are lines joining vertices.

One way to denote an edge e is to explicit the 2 vertices that are connected by this edge, say if the edge e links the vertex u to the vertex v, we write $e = \{u, v\}$.

Definition 73. Two vertices u, v in a graph G are adjacent in G if $\{u, v\}$ is an edge of G. If $e = \{u, v\}$ is an edge of G, then e is called incident with the vertices u and v.

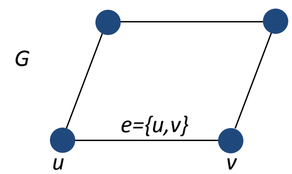
Graphs can be of several types.

Definition 74. A simple graph G is a graph that has no loop, that is no edge $\{u, v\}$ with $u = v$ and no parallel edges between any pair of vertices.

Example 106. Consider the table of fictitious flights among different cities. Suppose all you want to know is whether there is a direct flight between any two cities, and you are not interested in the direction of the flight. Then you can draw a graph whose vertices are the cities, and there is an edge between city A and city B exactly when there is at least one flight going either from city A to the city B, or from city B to city A.

Definitions

A graph G = (V,E) is a structure consisting of a set V of vertices (nodes) and a set E of edges (lines joining vertices).



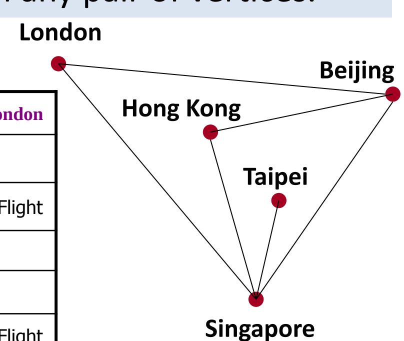
- Two vertices *u* and *v* are adjacent in *G* if {*u*,*v*} is an edge of *G*.
- If *e* = {*u*,*v*}, the edge ***e*** is called incident with the vertices *u* and *v*.

Graphs are useful to represent data.

Simple Graphs

A simple graph is a graph that has no loop (=edge {u,v} with u=v) and no parallel edges between any pair of vertices.

From\To	Hong Kong	Singapore	Beijing	Taipei	London
Hong Kong		4 Flights			
Singapore	2 Flights		3 Flights	1 Flight	1 Flight
Beijing	1 Flight	2 Flights			
Taipei					
London		1 Flight	1 Flight		1 Flight



Draw a graph to see whether there are direct flights between any two cities (in either direction)

Take for example Hong Kong: there will be one edge between Hong Kong and Singapore (this is read in the first row), and and one edge from Hong Kong to Beijing (this is read in the first column). This graph is simple, there is no loop, and no parallel edge.

Definition 75. A multigraph G is a graph that has no loop and at least two parallel edges between some pair of vertices.

Example 107. We continue with our fictitious example of flights. Suppose now we want to know how many flights are there that operate between two cities (we are still not interested in the direction). In this case, we will need parallel edges to represent multiple flights. For example, let us consider Hong Kong and Singapore: there are 4 flights from Hong Kong to Singapore, and 2 flights from Singapore to Hong Kong, thus a total of 6 edges between the two vertices representing these cities.

Very often, a relation from a vertex A to B does not yield one from B to A, in this case, edges become arrows.

Definition 76. A directed graph G, also called digraph for short, is a graph where edges $\{u, v\}$ are ordered ($\{u, v\}$ and $\{v, u\}$ are not the same), that is edges have a direction. The graph is called undirected otherwise. Parallel edges are allowed in directed multigraph. Loops are allowed for both directed and directed multigraphs.

Example 108. On our example of ficticious flights, a directed graph corresponds to put arrows for flights going in a given direction, for example, there is one arrow from London to Beijing, but none from Beijing to London.

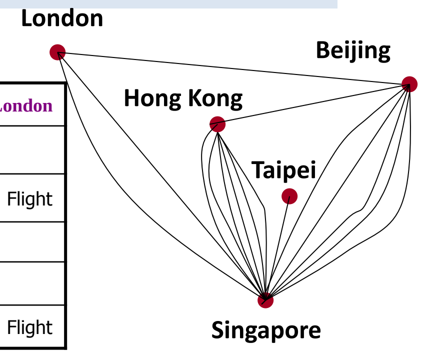
People attribute the origin of graph theory to the Königsberg bridge question, solved by the mathematician Euler in 1736. The question was, given the map of Königsberg, which contains 7 bridges, is it possible to find a walk that goes through the 7 bridges without crossing a bridge twice? You may look at the map and give it a try yourself, to convince yourself of the answer. The answer turns out to be no, it is not possible. We will see why next. But first, we will give a name to such walks, in honour of Euler:

Definition 77. A Euler path/trail is a walk on the edges of a graph which uses each edge in the graph exactly once. A Euler circuit/cycle is a walk on the edges of a graph which starts and ends at the same vertex, and uses each edge in the graph exactly once.

Multigraphs

A multigraph is a graph that has no loop and at least 2 parallel edges between some pair of vertices.

From\To	Hong Kong	Singapore	Beijing	Taipei	London
Hong Kong		4 Flights			
Singapore	2 Flights		3 Flights	1 Flight	1 Flight
Beijing	1 Flight	2 Flights			
Taipei					
London		1 Flight	1 Flight		1 Flight

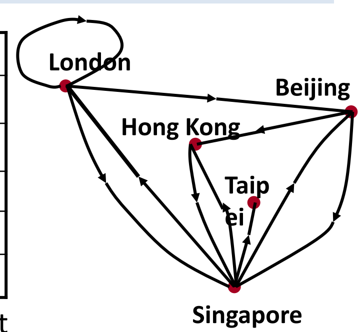


Draw a graph with an edge for each flight that operates between two cities (in either direction).

Directed (Multi)graphs

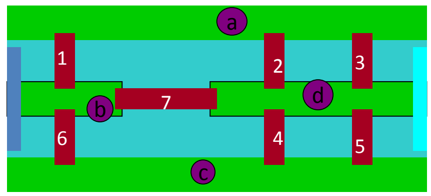
A directed graph is a graph where edges {u,v} are ordered, that is, edges have a direction. Parallel edges are allowed in directed multigraphs. Loops are allowed for both.

From\To	Hong Kong	Singapore	Beijing	Taipei	London
Hong Kong		4 Flights			
Singapore	2 Flights		3 Flights	1 Flight	1 Flight
Beijing	1 Flight	2 Flights			
Taipei					
London		1 Flight	1 Flight		1 Flight

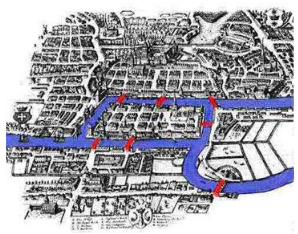


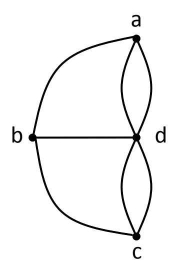
Draw a graph to see whether there are direct flights between any two cities (direction matters)

Origin: The Bridges of Konigsberg



- Königsberg (now Kaliningrad, Russia) has 7 bridges.





Leonhard Euler introduced Graphs in 1736 to solve the Königsberg Bridge problem (no solution and why).

- People tried (without success) to find a way to walk all 7 bridges without crossing a bridge twice.

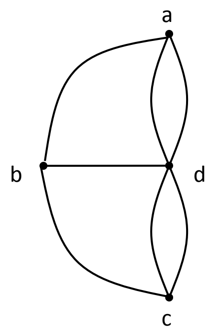
Euler Path and Circuit

A Euler path (Eulerian trail) is a walk on the edges of a graph which uses each edge in the original graph exactly once.

The beginning and end of the walk may or not be the same vertex.

A Euler circuit (Eulerian cycle) is a walk on the edges of a graph which starts and ends at the same vertex, and uses each edge in the original graph exactly once.

Euler Circuit



- Suppose the beginning and end are the same node u.
- The graph must be connected.
- At every vertex v≠ u, we reach v along one edge and go out along another, thus the number of edges incident at v (called the degree of v) is even.
- The node u is visited once the first time we leave, and once the last time we arrive, and possibly in between (back and forth), thus the degree of u is even.
- Since the Königsberg Bridges graph has odd degrees, no solution!

8/15

Euler Theorem

The degree of a vertex is the number of edges incident with it.

Theorem. Consider a connected graph G.

1. If G contains an Euler path that starts and ends at the same node, then all nodes of G have an even degree.
2. If G contains an Euler path, then exactly two nodes of G have an odd degree.

- Suppose G as an Euler path, which starts at v and finishes at w.
- Add the edge {v,w}.
- Then by the first part of the theorem, all nodes have even degree, but for v and w which have odd degrees.

The Euler circuit/cycle is simply an Euler path/trail whose start and end are the same vertex.

Definition 78. The degree of a node is the number of edges incident with it.

With this definition of degree, we next answer the question of the bridges of Kñigsberg. We start with the more constrained case of starting and finishing at the same vertex. We assume the graph G is connected, which means that there is always a way to walk from any vertex to any other (possibly using several times the same vertex or the same edge).

Theorem 4. *Consider a connected graph G. Then G contains an Euler circuit/cycle, if and only if all nodes of G have an even degree.*

Proof. We prove only that if G contains an Euler circuit, then all nodes have an even degree. We start the walk at vertex u. Now for any vertex v which is not u, we need to walk in v using some edge, and walk out of u using another edge. We may come back to v, but for every come back, we still need one edge to come in, and one to walk out. Therefore the degree of v must be even! As for the starting point u, it is visited once the first time we leave, and the last time we arrive (2 edges), and any possible back and forth counts for 2 edges as well, which shows that indeed, it must be that all nodes have an even degree! □

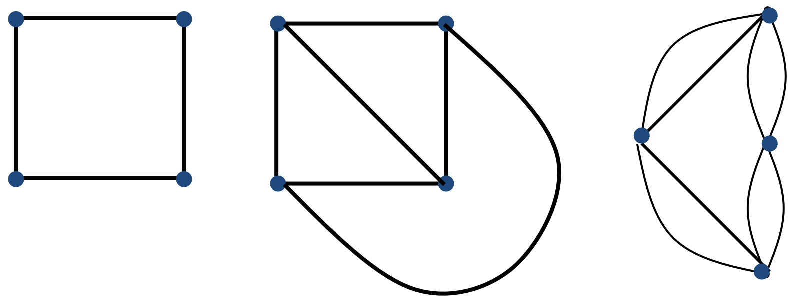
Since the Königsberg bridge graph has odd degrees, it has no Euler cycle. We next extend the argument to an Euler path.

Theorem 5. *Consider a connected graph G. Then G contains an Euler path if and only if exactly two nodes of G have an odd degree.*

Proof. We only prove that if G has an Euler path, then exactly two nodes of G have an odd degree. Suppose thus that G has an Euler path, which starts at v and finishes at w. Create a new graph G', which is formed from G by adding one edge between v and w. Now G' has an Euler cycle, and so we know by the previoust theorem that G' has the property that all its vertices have an even degree. Therefore the degrees of v and w in G are odd, while all the others are even and we are done. The other direction is left as an exercise (see Exercise 96). □

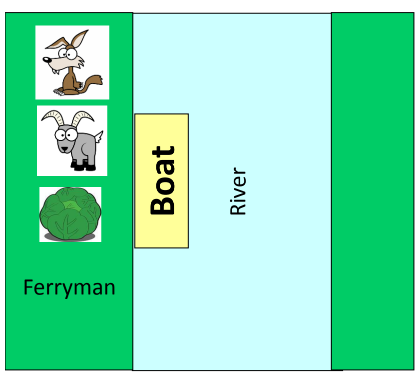
Examples

Note: Euler Theorem actually states an if and only if.



Example: Wolf, Goat, Cabbage

A classical puzzle that involves graphs:



From the left bank of the river, the ferryman is to transport the wolf, the goat and the cabbage to the right bank.

The boat is only big enough to transport one object/animal at a time, other than himself.

The wolf cannot be left alone with the goat, and the goat cannot be left alone with the cabbage.

How should the ferryman proceed?

Here is a classical puzzle.

Example 109. A ferryman needs to transport a wolf, a goat and a cabbage from one side of a river to the other, the boat is big enough for himself and one object/animal at a time. How should the ferryman proceed, knowing that the wolf cannot be left alone with the goat, and the goat cannot be left alone with the cabbage? One way to solve this is to use a graph that represents the different possible states of this sytem. The initial start up point is a state where $wgcf$ (w=wolf, g=goat, c=cabbage and f=ferryman) are on the left side of the river, with nothing on the right side. The first step, the ferryman has no choice, he takes the goat on the other side, which leads to a second step, with wc on the leftt bank, and gf on the right bank. The ferryman returns, he then can choose: he either takes the cabbage of the wolf. This creates two branches in the graph. Each branch leads to a couple of states, after which (see the graph itself) we reach a state where g is on the left, and wfc is on the right, leading to the end of the puzzle. A solution is a path in this graph that represents the different states of the system. In this example, it is a fairly easy graph, and there are 2 paths, each of the same length!

Here are two more types of graphs which are important, in that you are very likely to encounter them.

Definition 79. A complete graph with n vertices is a simple graph that has every vertex connected to every other distinct vertex.

Definition 80. A bipartite graph is a graph whose vertices can be partitioned into 2 (disjoint) subset V and W such that each edge only connects a $v \in V$ and a $w \in W$.

We have seen the notion of degree of a vertex v in an undirected graph, it is the number of edges incident with. We note that in this case, a loop at a vertex contributes twice. For directed graphs, you may like to know that the notion of degree is more precise, one distinguishes in-degree and out-degree, which we will not discussed here.

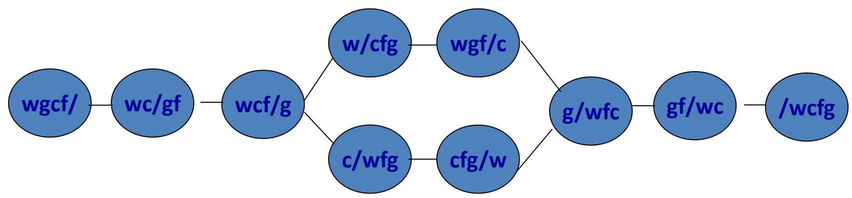
Definition 81. The total degree of an undirected graph $G = (V, E)$ is the sum of the degrees of all the vertices of G: $\sum_{v \in V} \deg(v)$.

Example: Wolf, Goat, Cabbage (IV)

- f = ferryman
- g = goat
- w = wolf
- c = cabbage

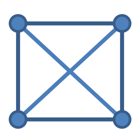
Either he takes

1. The cabbage, bring back the goat, leave the goat and take the wolf across, return, and take the goat across.
2. Or the wolf, bring back the goat, leave the goat and the cabbage across, return, and take the goat across.

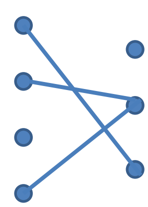


Complete & Bipartite

A complete graph with n vertices is a simple graph that has every vertex connected to every other distinct vertex.



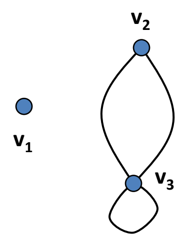
A bipartite graph is a graph whose vertices can be partitioned into 2 (disjoint) subsets V and W s.t. each edge only connects a $v \in V$ and a $w \in W$.



More on Node Degree

The degree deg(v) of a vertex *v* in an undirected graph is the number of edges incident with it (a loop at a vertex contributes twice) .
In-degree and Out-degree are distinguished for directed graphs.

total degree = deg(v_1) + deg(v_2) + deg(v_3) = 0 + 2 + 4 = **6**.



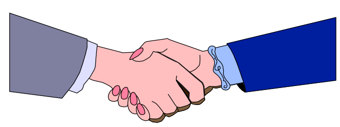
The total degree deg(G) of an undirected graph G is the sum of the degrees of all the vertices of *G:* $\sum_{v \in V} \deg(v)$

The Handshaking Theorem

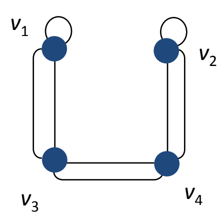
Let G = (V,E) be an undirected graph with e edges. Then

$$2e = \sum_{v \in V} \deg(v)$$

(Note that this even applies if multiple edges and loops are present.)

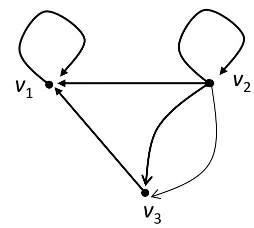


Proof.
Choose an e ∈ E(G) with endpoints v, w ∈ V.
e contributes 1 to deg(v) and 1 to deg(w). True even when v = w . Thus each edge contributes 2 to the total degree.



deg(v_1) = deg(v_2) = deg(v_3) = deg(v_4) = 4
$2e = \sum \deg(v) = 4 \times 4 = 16$ and $\boldsymbol{e = 8}$

Adjacency Matrix



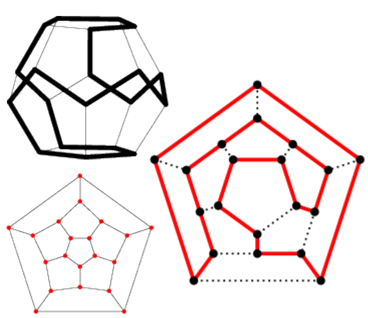
$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

A graph can be represented by a matrix $A = (a_{ij})$ called adjacency matrix, with

a_{ij} = the number of arrows from v_i to v_j

What is the adjacency matrix of a complete graph?

Hamiltonian Circuit



- *The Icosian game* (1857):
Along the edges of a dodecahedron, find a path such that every vertex is visited a single time, and the ending point is the same as the starting point.
- Hamilton sold it to a London game dealer in 1859 for 25 pounds.

A Hamiltonian path of a graph G is a walk such that every vertex is visited exactly once.

A Hamiltonian circuit of a graph G is a closed walk such that every vertex is visited exactly once (except the same start/end vertex).

The Handshaking Theorem links the number of egdes in a graph to the numbers of vertices.

Theorem 6. *Let $G = (V, E)$ be an undirected graph with $|E|$ edges. Then*

$$2|E| = \sum_{v \in V} \deg(v).$$

Proof. The proof follows the name of the theorem. The idea of handshaking is that if two people shake hands, there must be...well...two persons involved. In a graph G, this becomes, if there is an edge e between v and w, then e contributes to 1 to the degree of v and to 1 to the degree of s. This is also true when $v = w$. Therefore each edge contributes 2 to the total degree. $\square$

To represent a graph, a useful way to do so is to use a matrix.

Definition 82. The adjacency matrix of a graph G is a matrix A whose coefficients are denoted a_{ij}, where a_{ij}, the coefficient in the ith row and jth column, counts the number of arrows from v_i to v_j.

You may replace the term arrow by edge in this definition if you graph is undirected.

Example 110. The adjacency of a complete graph with 4 vertices is

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

The first row reads that v_1 is connected to v_2, v_3, v_4, but not to itself (since it is a simple graph by definition).

We saw earlier Euler paths as walks going through exactly every edge of a graph. If you replace "exactly every edge" by "exactly every vertex", this becomes a Hamiltonian path!

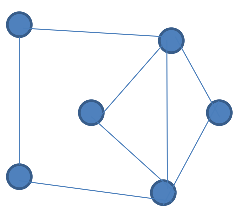
Definition 83. A Hamiltonian path of a graph G is a walk such that every vertex is visited exactly one. A Hamiltonian circuit of a graph G is a closed walk such that every vertex is visited exactly one, except the same start/end vertex.

Hamiltonian paths are harder to characterize than Euler paths. In particular, finding an algorithm that will identify Hamiltonian paths in a graph is hard!

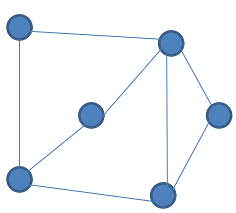
Hamiltonian vs Eulerian

- Path (or trail) vs Circuit (or cycle): for circuits, the walk starts and finishes at the same vertex, not needed for path.
- Eulerian: walk through every edge exactly once.
- Hamiltonian: walk through every vertex exactly.

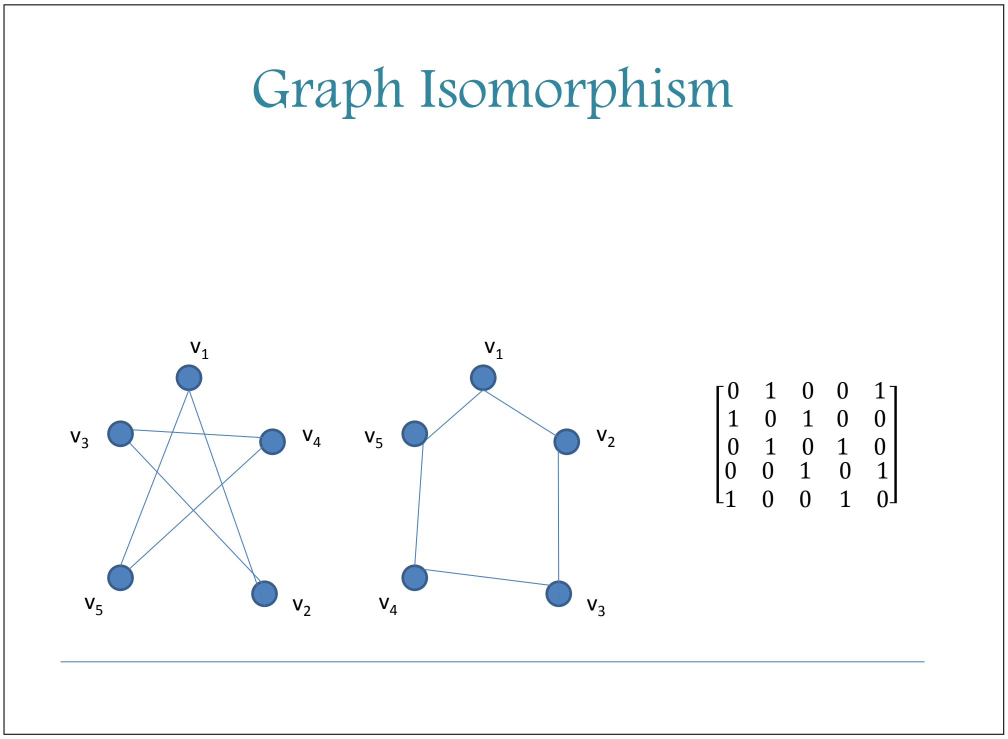
Examples



Euler circuit
Hamiltonian path
No Hamiltonian circuit



No Euler circuit, but Euler path
Hamiltonian path
No Hamiltonian circuit



Finally, it is useful to pay attention that one draws a graph, it is just a visualization...and several visualizations may be different, giving the impression that the graphs are different, while they are actually the same. If two graphs differ by their labeling, but their adjacency structure is the same, we say that these graphs are isomorphic. More formally:

Definition 84. A graph isomorphism between two graphs G and H is a bijection f between the set of vertices of G and the set of vertices of H such that any two vertices u, v in G are adjacent if and only if $f(u), f(v)$ are adjacent in H.

Exercises for Chapter 11

Exercise 96. Prove that if a connected graph G has exactly two vertices which have odd degree, then it contains an Euler path.

Exercise 97. Draw a complete graph with 5 vertices.

Exercise 98. Show that in every graph G, the number of vertices of odd degree is even.

Exercise 99. Show that in very simple graph (with at least two vertices), there must be two vertices that have the same degree.

Exercise 100. Decide whether the following graphs contain a Euler path/cycle.