Graphs

Def. A simple graph is a pair of sets $G = (V, E)$ where
$E \subseteq \{\{a,b\} : a,b \in V,\ a \neq b\}$.

Recall how we used graphs for relations. (We did use directed graphs there though — basically (a,b) instead of $\{a,b\}$).

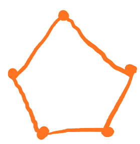
A simple graph is basically a symmetric irreflexive relation.
($\{a,b\}$ is modelled as (a,b) and (b,a))

In a non-simple graph, we allow more than one edge between a pair of nodes (multigraph) or more generally, we label edges with weights.

The complete graph K_n is the graph with n nodes and all possible edges between them.
($E = \{\{a,b\} : a,b \in V \text{ and } a \neq b\}$)

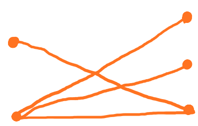
K_5

A cycle C_n is a graph with $V = \{v_1, v_2, \ldots, v_n\}$ and
$E = \{\{v_i, v_j\} : j = i+1 \text{ or } i = n \text{ and } j = 1\}$



C_5

A graph is said to be bipartite if there exist non-empty disjoint sets V_1 and V_2 such that $V = V_1 \cup V_2$ and
$E \subseteq \{\{a,b\} : a \in V_1 \text{ and } b \in V_2\}$. (there are no edges within a part)



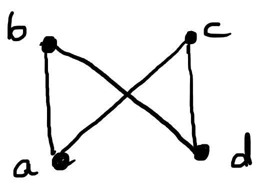
For even n, C_n is bipartite

A complete bipartite graph K_{n_1,n_2} is a bipartite graph with $|V_1| = n_1$, $|V_2| = n_2$, and $E = \{\{a,b\} : a \in V_1, b \in V_2\}$.
$(|E| = n_1 n_2)$

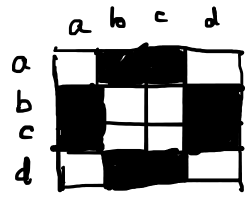
Def. Graphs $G_1 = (E_1, V_1)$ and $G_2 = (E_2, V_2)$ are *isomorphic* if there is a bijection $f: V_1 \to V_2$ such that $\{u,v\} \in E_1$ iff $\{f(u), f(v)\} \in E_2$.

(They have the same "structure")

We can also describe graphs by adjacency matrices:



is



Then two graphs are isomorphic if we can permute the rows and columns (in the same sense) of one matrix to get the other.

A computational problem is to determine if two graphs are isomorphic from their adjacency matrices.
There is no general efficient algorithm known for the above for large graphs.

Def. A *subgraph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$.

To get a subgraph,

1. Remove zero or more vertices along with the edges incident on them.
2. Further remove zero or more edges.

We get an *induced subgraph* by omitting the latter step.

Walks and Paths

Def. A walk of length $k \geq 0$ from node a to node b is a sequence of nodes $(a = v_0, v_1, \ldots, v_k = b)$ such that for all $0 \leq i \leq k-1$, $\{v_i, v_{i+1}\} \in E$.

If a walk has no repeating nodes, it is a path.

If a walk of length $k \geq 3$ has $v_0 = v_k$ and has no other repeating nodes, it is called a cycle.
A graph is acyclic if it has no cycles (there is no subgraph isomorphic to C_k)

Def. Nodes u and v are said to be connected if there exists a path from u to v.
Equivalently, they are connected if there is a walk between them. (Why?)

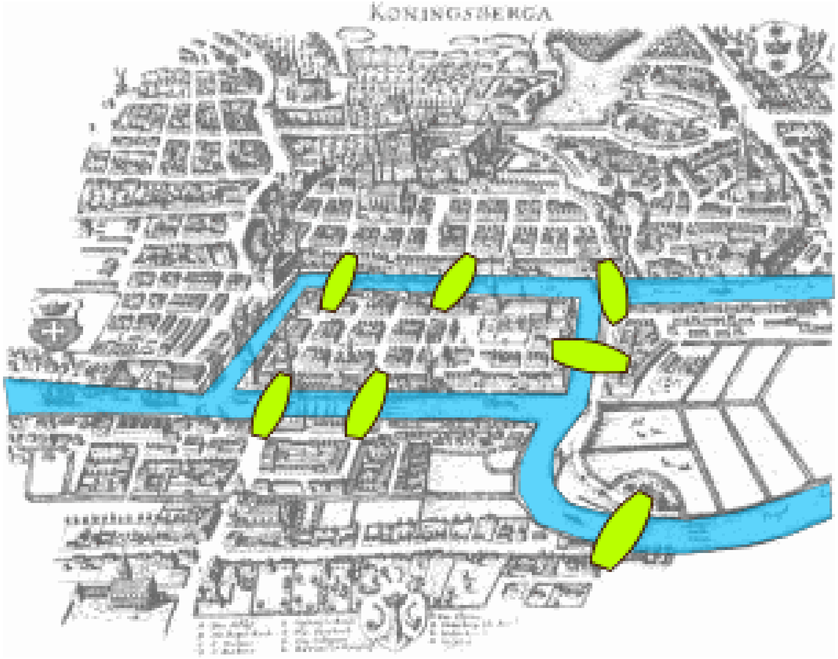
The connectedness relation is an equivalence relation.
The equivalence classes of this relation are called the connected components of G.

Given a simple graph $G = (V, E)$, the degree of $v \in V$ is the number of edges incident on v. That is,

$$\deg(v) = |\{u : \{u, v\} \in E\}|$$

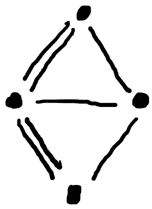
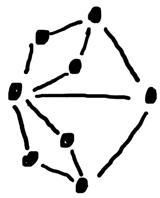
Note that $2|E| = \sum_{v \in V} \deg(v)$ (each edge is counted twice)

The degree sequence of a graph is a sorted list of degrees.
It is invariant under isomorphism.



A famous question known as the "Seven Bridges of Königsberg" asks whether it is possible to walk through the city crossing each bridge exactly once.

We can model this as a graph:

 or equivalently, as a simple graph: 

With this motivation, define an Eulerian trail as a walk visiting each edge exactly once.

The question then asks if an Eulerian trail exists for the latter graph.

Note that if an Eulerian trail exists, there must be at most 2 odd degree nodes.

Indeed, define

$$\mathrm{Enter}(v) = \{\{v_{i-1}, v_i\} : v_i = v\}$$
$$\mathrm{Exit}(v) = \{\{v_i, v_{i+1}\} : v_i = v\}$$

that partitions all edges incident on v. Further, $|\mathrm{Enter}(v)| = |\mathrm{Exit}(v)|$ for all v except the start and end nodes.

$\Rightarrow$ There can be at most two odd degree nodes.

(Namely, the start and end)

As the graph drawn above has 3 odd degree nodes no Eulerian walk exists.

An Eulerian circuit is a closed walk ($v_0 = v_k$) visiting each edge exactly once.
If an Eulerian circuit exists, there are no odd degree nodes.

Further, if there are no odd degree nodes and all edges appear in a single connected component, there must exist an Eulerian circuit!
(Try splitting it into several cycles and "stitching" them together)
This also gives an efficient algorithm to find an Eulerian circuit if it exists.

A Hamiltonian cycle is a cycle that contains all nodes in the graph.

There is no efficient algorithm known to check if a graph has a Hamiltonian cycle.
Indeed, this is an "NP-hard" problem.

Given connected nodes u and v, the distance between them is the length of a shortest walk between them. (and ∞ if no walk exists)

this shortest walk must be a path.

We also define the diameter of a graph as the largest distance between two nodes in a graph.
(This is ∞ if the graph is not connected)