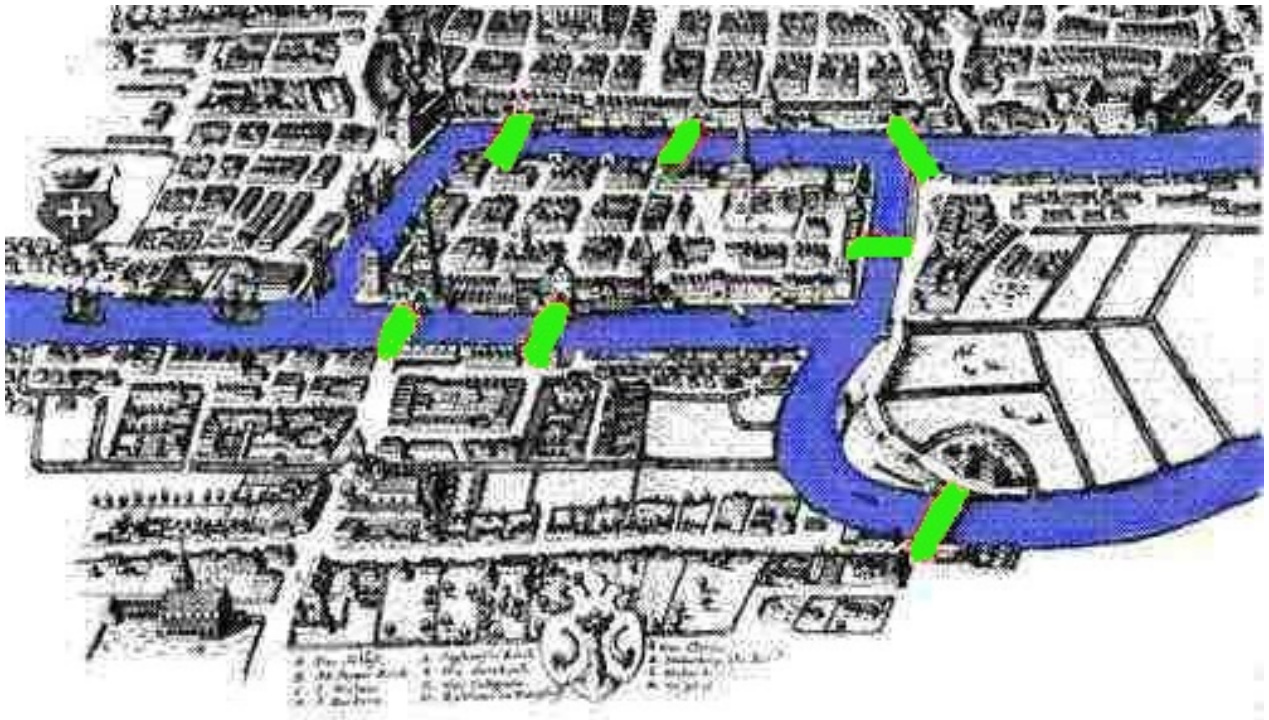
The Bridges of Königsberg
COMS20017 (Algorithms and Data)

John Lapinskas, University of Bristol

The eponymous bridges



The residents of Königsberg in the 18^{th} century had a game: they would try to walk through the city while crossing every bridge exactly once.

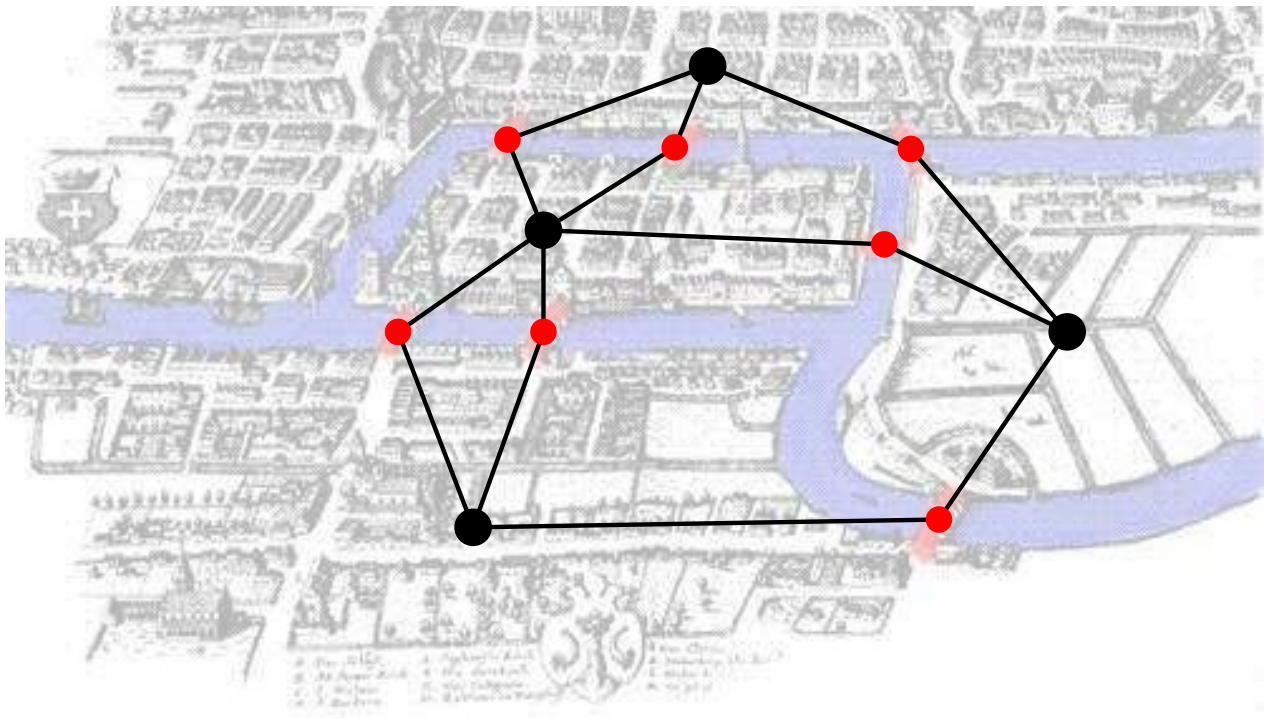
In 1736, the mayor of Danzig asked Leonhard Euler whether or not it was actually possible, inadvertently laying the foundations for an entire branch of mathematics.

Basic definitions

A **graph** is a pair $G = (V, E)$, where $V = \boldsymbol{V(G)}$ is a set of **vertices** and $E = \boldsymbol{E(G)}$ is a set of **edges** contained in $\{\{u, v\} : u, v \in V,\ u \neq v\}$.

We picture vertices as dots, and each edge $\{u, v\}$ as a line joining u and v.

For example, if our vertices are islands and bridges, and we join an island to a bridge if they're adjacent:

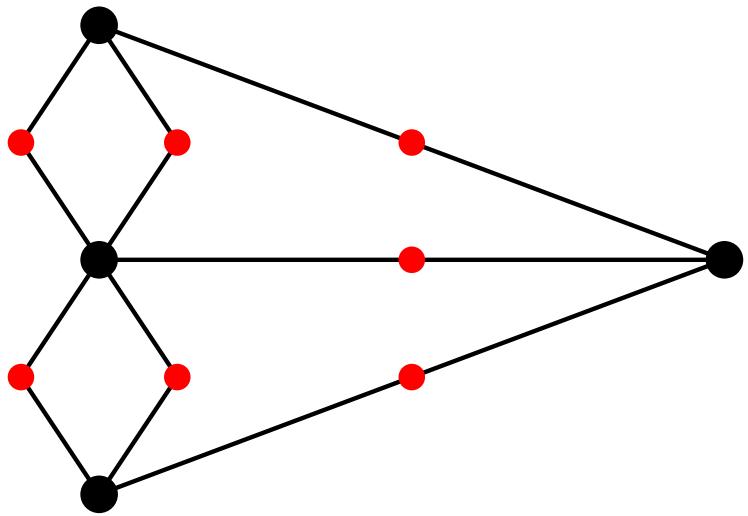


Basic definitions

A **graph** is a pair $G = (V, E)$, where $V = \boldsymbol{V(G)}$ is a set of **vertices** and $E = \boldsymbol{E(G)}$ is a set of **edges** contained in $\{\{u, v\} : u, v \in V,\ u \neq v\}$.

We picture vertices as dots, and each edge $\{u, v\}$ as a line joining u and v.

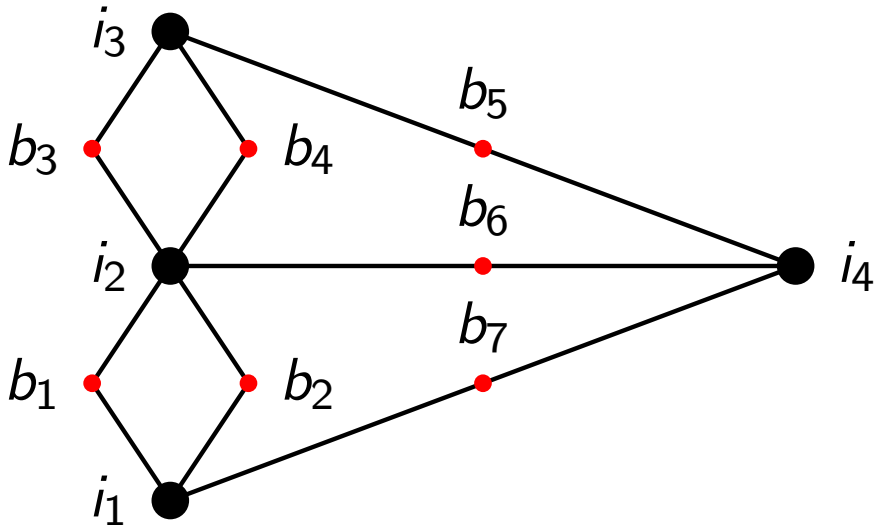
For example, if our vertices are islands and bridges, and we join an island to a bridge if they're adjacent:



A **walk** in a graph $G = (V, E)$ is a sequence of vertices $v_0 \dots v_k$ such that $\{v_i, v_{i+1}\} \in E$ for all $i \leq k - 1$.

We say the walk is **from** v_0 **to** v_k and call k the **length** of the walk.

For example: $i_1\ b_1\ i_2\ b_3\ i_3\ b_4\ i_2\ b_2\ i_1\ b_7\ i_4\ b_6\ i_2$.



An **Euler walk** is one which contains every edge in G exactly once.
The mayor effectively asked: does this graph have an Euler walk?

Why bother?

For this specific problem, we could just think about Königsberg itself, or about general islands and bridges. (That's what Euler did!)

But using general graphs comes with three advantages:

- The graph picture is **easier** to think about than a picture of Königsberg — the irrelevant detail is stripped away.
- By learning about graphs, we can easily **recognise** when real-world problems are actually graph problems we already know how to solve. (Like using design patterns rather than reinventing the wheel.)
- By building up a general theory of graphs, rather than starting from scratch for each individual problem, we can solve far **harder** problems that we would otherwise be able to.

One disadvantage: The next few videos will have a lot of definitions!

But it's just a toy problem!

Graphs can be used to model far more things than just Königsberg!

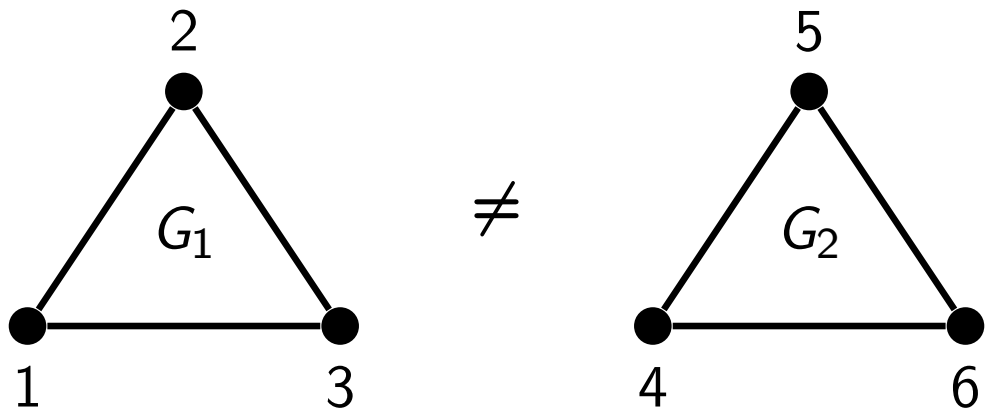
- The National Grid.
- The road system in a city.
- The Internet.
- Social networks like Facebook.
- Resource allocation problems.
- Circuit design problems.
- Dependencies between software packages.
- Evolutionary relationships between organisms.
- The spread of an epidemic.
- Route planning problems.
- Heap data structures as seen in COMS10007.

And many, many, many more...

When are two graphs equal?

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **equal**, written $G_1 = G_2$, if $V_1 = V_2$ and $E_1 = E_2$.

But that means...



even though they're clearly "the same graph"!

G_1 and G_2 are **isomorphic**, written $G_1 \simeq G_2$, if there's a bijection $f : V_1 \to V_2$ such that $\{f(u), f(v)\} \in E_2$ if and only if $\{u, v\} \in E_1$.

Intuitively: $G_1 \simeq G_2$ if they are the same graph with relabelled vertices. Here, $G_1 \neq G_2$ but $G_1 \simeq G_2$. (Take e.g. $f(1) = 4$, $f(2) = 5$, $f(3) = 6$.)

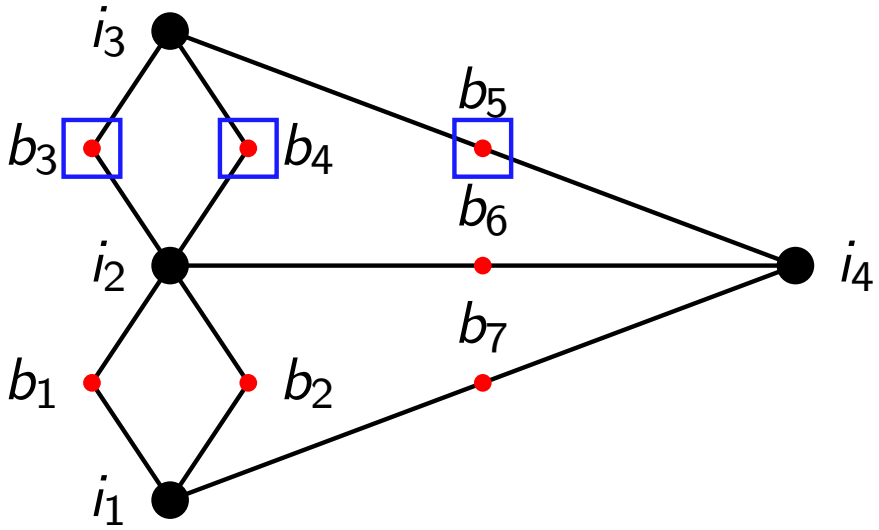
Another terminology slide...

In a graph $G = (V, E)$, the **neighbourhood** of a vertex v is the **set** of vertices joined to v by an edge. Formally: $N_G(v) = \{w \in V : \{v, w\} \in E\}$.

We just write this as $N(v)$ if G is obvious from context. Also, for all sets of vertices $X \subseteq V$, we write $N_G(X) = \bigcup_{v \in X} N_G(v)$.

The **degree** of a vertex v is the **number** of vertices joined to v. Formally: $d_G(v) = |N_G(v)|$.

Again, we just write this as $d(v)$ if G is obvious from context.



Here $N(i_3) = \{b_3, b_4, b_5\}$, and $d(i_3) = 3$.

Back to the bridges!

A **walk** in a graph $G = (V, E)$ is a sequence of vertices $v_0 \dots v_k$ such that $\{v_i, v_{i+1}\} \in E$ for all $i \leq k - 1$. An **Euler walk** is one which contains every edge in G exactly once.

The **degree** of v, $d(v)$, is the number of vertices joined to v by edges.

Euler noticed: any walk with $v_0 = v_k$ uses an **even** number of edges from every vertex, since it leaves each vertex immediately after entering.

Similarly, any walk with $v_0 \neq v_k$ uses an **odd** number of edges from v_0 and v_k, and an **even** number from any other vertex.

But an Euler walk has to use every edge at every vertex!

Theorem: If G has an Euler walk, then either:

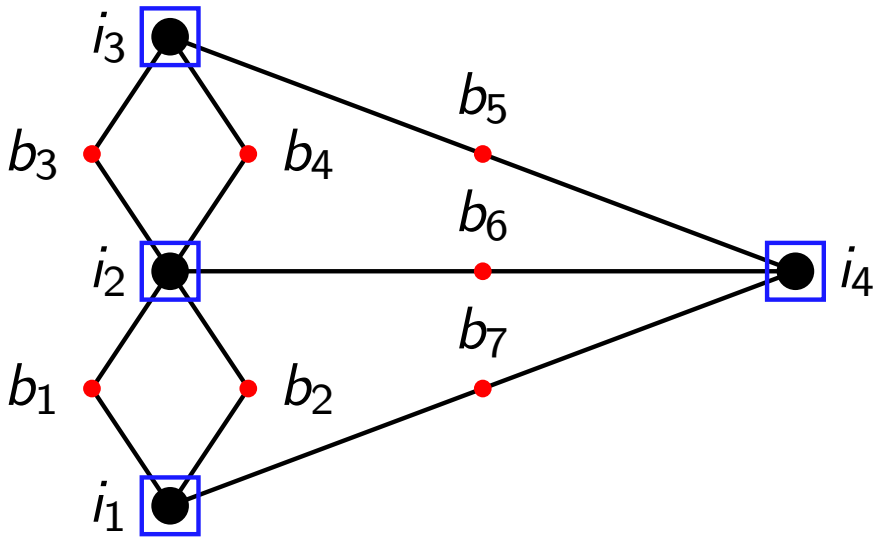
- every vertex of G has even degree; or
- all but two vertices v_0 and v_k have even degree, and any Euler walk must have v_0 and v_k as endpoints. □

A **walk** in a graph $G = (V, E)$ is a sequence of vertices $v_0 \dots v_k$ such that $\{v_i, v_{i+1}\} \in E$ for all $i \leq k - 1$. An **Euler walk** is one which contains every edge in G exactly once.

The **degree** of v, $d(v)$, is the number of vertices joined to v by edges.

Theorem: If G has an Euler walk, then either:

- every vertex of G has even degree; or
- all but two vertices v_0 and v_k have even degree, and any Euler walk must have v_0 and v_k as endpoints.



In Königsberg, i_1, i_2, i_3 and i_4 **all** have odd degree, so no Euler walk exists!