The Origins of Graph Theory

City of Königsberg in 1736

Is it possible to start out anywhere, cross all seven bridges exactly once, and return to where you started?

We can model this situation with a graph:

Question. Can we draw this graph without lifting our pencil?

Pseudographs

This is not a graph—it's a pseudograph. For this section and others, we will allow *multiple edges* and *loops*.

A few of our definitions need to be updated.

Types of "walks" in pseudographs:

Repeat Vertices?	Repeat Edges?	Open $A_1 \neq A_n$	Closed $A_1 = A_n$
No	No	path	cycle
Yes	No	trail	circuit
Yes	Yes	walk	closed walk

Definition: The **length** of a "walk" is the number of *edges* involved.

Remark. In a simple graph, the smallest cycle possible is of length 3. In a pseudograph, there may exist cycles of length 1 and 2.

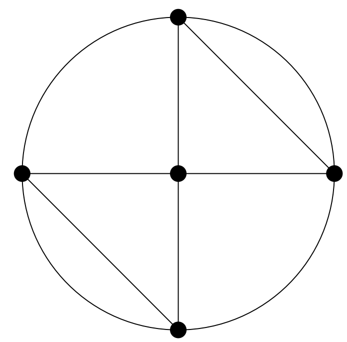
Definition: The **degree** of a vertex v is the number of edges incident with v; loops count twice!

Eulerian Circuits

Definition: An **Eulerian circuit** C in a graph G is a circuit containing every edge of G.
Definition: An **Eulerian trail** T in a graph G is a trail containing every edge of G.

A graph with an Eulerian circuit does not have an Eulerian trail.

⋆ Important: Trails and circuits do not use any edge twice. ⋆



So the Königsberg bridge problem in the language of graph theory is:

Is there an Eulerian circuit in the corresponding pseudograph?

Characterization of Graphs with Eulerian Circuits

There is a simple way to determine if a graph has an Eulerian circuit.

Theorems 3.1.1 and 3.1.2. Let G be a pseudograph that is connected* *except possibly for isolated vertices*. Then, G has an Eulerian circuit $\iff$ the degree of every vertex is even.

The Königsberg bridge pseudograph has four vertices of odd degree, and therefore does not have an Eulerian circuit.

($\Rightarrow$) Euler, 1736. Given an Eulerian circuit C, each time a vertex appears in the circuit, there must be an "in edge" and an "out edge", so the total degree of each vertex must be even.

($\Leftarrow$) Hierholzer, 1873. This is harder; we need the following lemma.

Proof of Lemma 3.1.3

Lemma 3.1.3. If the degree of every vertex in a pseudograph is even, then every non-isolated vertex lies in some circuit in G.

Proof. Start a trail at any non-isolated vertex A in G.

Whenever the trail arrives at some other vertex B, there must be an odd number of edges incident to B not yet traversed by the trail.

So there is some edge to follow out of B; take it.

The trail must eventually return to A, giving us a circuit.

Proof of Theorem 3.1.2

$\star$ Each vertex in G has even degree $\Rightarrow$ G has an Eulerian circuit $\star$

Find the longest circuit C in G. If C uses every edge, we are done. If not, we'll show a contradiction to the maximality of C.

Remove all edges of C from G and any isolated vertices to form H. H is a pseudograph where each vertex is of even degree.

Since G is connected, C and H must share a vertex A.
Write C as $C = \cdots e_1 A e_2 \cdots$.

Find a circuit D in H through A.
Write D as $D = \cdots f_1 A f_2 \cdots$.
No edges of D repeat nor are they in C.

Define a new circuit $C' = \cdots e_1 A f_2 \cdots f_1 A e_2 \cdots$.

C' is a longer circuit in G than C, contradicting C's maximality. $\square$

Other related theorems

Theorem 3.1.6. Let G be a connected* pseudograph. Then, G has an Eulerian trail $\Leftrightarrow$ G has exactly two vertices of odd degree.

Proof. Let x and y be the two vertices of odd degree. Add edge xy to G; $G + xy$ is a pseudograph with each vertex of even degree. By Theorem 3.1.2, there exists an Eulerian circuit in $G + xy$. Remove xy from the circuit and you have an Eulerian trail in G.

Remark. When drawing a picture without lifting your pencil, start and end at the vertices of odd degree!

Theorem 3.1.5. A pseudograph G has a decomposition into cycles if and only if every vertex has even degree.

Application: de Bruijn sequences

Consider the following example of a **de Bruijn sequence**:

$$0000110101111001$$

Each of the sixteen binary sequences of length 4 are present (where we allow cycling):

0000	0100	1000	1100
0001	0101	1001	1101
0010	0110	1010	1110
0011	0111	1011	1111

This is the most compact way to represent these sixteen sequences.

Sequence definitions

Definition: An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition: A **sequence** or **word** from $\mathcal{A}$ is a succession $S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition: A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Definition: A **de Bruijn sequence** of order n on the alphabet $\mathcal{A}$ is a sequence of length k^n such that every word of length n is a consecutive subsequence of S. (and called **binary** if $\mathcal{A} = \{0, 1\}$)

Theorem. A de Bruijn sequence of any order n on any alphabet $\mathcal{A}$ always exists.

Proof. Use the theory of Eulerian circuits on certain graphs:

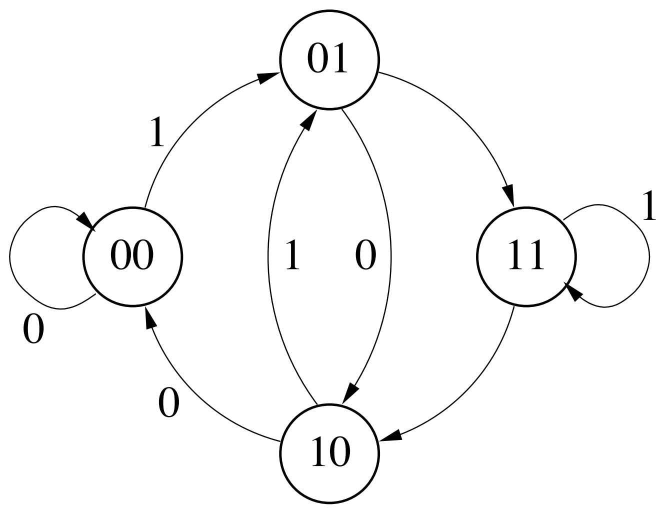
de Bruijn graphs

Definition: The **de Bruijn graph** of order n on $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph that has as its vertices words of $\mathcal{A}$ of length $n-1$. Each vertex has k out-edges corresponding to the k letters of the alphabet $\mathcal{A}$. Following edge a_i adds letter a_i to the end of the sequence and removes the first letter from the sequence:

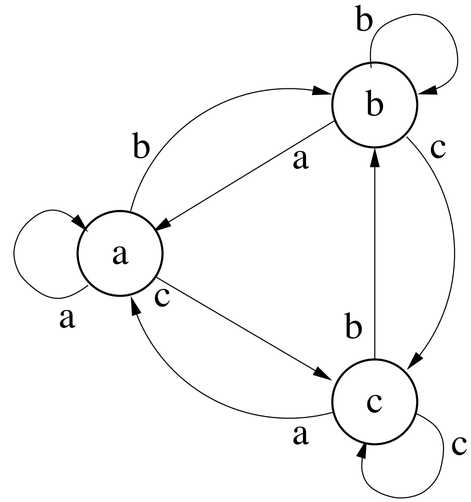
$$\boxed{b_1 b_2 \cdots b_{n-1}} \xrightarrow{a_i} \boxed{b_2 \cdots b_{n-1} a_i}$$

Examples.

The binary de Bruijn graph of order 3



The de Bruijn graph of order 2 on the alphabet $\mathcal{A} = \{a, b, c\}$.



Proof that a de Bruijn sequence always exists

The de Bruijn graph G of order n on alphabet $\mathcal{A}$ is connected and each vertex has as many edges entering as leaving the vertex. This implies that G has an Eulerian circuit C (of length k^n).

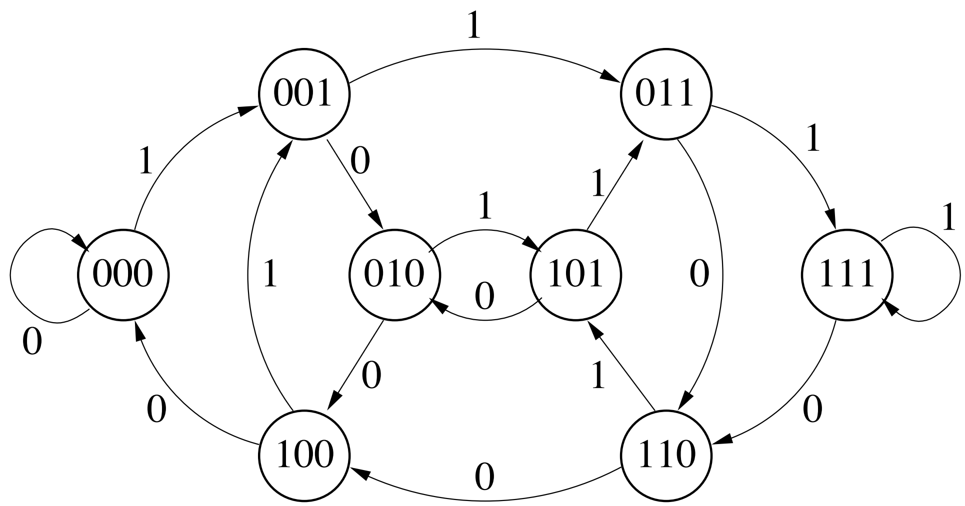
Follow C and record in order the sequence S of labels of edges visited.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

We know that S is of length k^n. We must now verify that every sequence of length n appears as a consecutive subsequence in S.

By construction, the sequence of the $n-1$ labels of edges visited before arriving at a vertex is exactly the name of the vertex. The word formed by this name followed by the label of an outgoing edge is a word of $\mathcal{A}$ of length n and is different for every edge of C. This implies that every sequence appears as a consecutive subseq. of S.

Example: The binary de Bruijn graph of order 4



1. Find an Eulerian circuit in this graph.
2. Write down the corresponding sequence.
3. Verify that it is a de Bruijn sequence. (use chart, p.66)
4. Convince yourself that the name of a vertex is the same as the sequence formed by the three previous edges.