Paths and Circuits

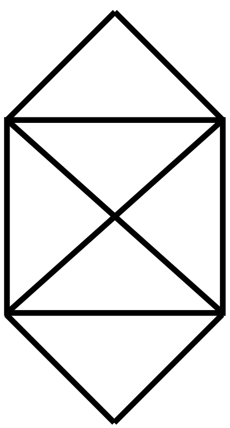
CSE 373 - Data Structures

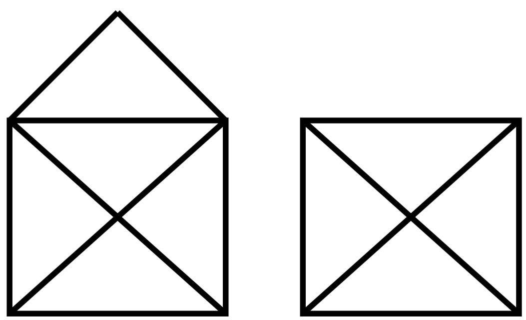
June 3, 2002

Readings and References

- Reading
  - › Section 9.6-9.7, *Data Structures and Algorithm Analysis in C*, Weiss
- Other References

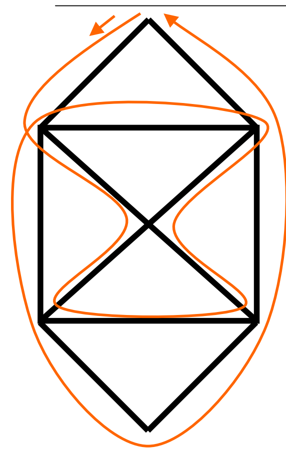
It’s Puzzle Time!

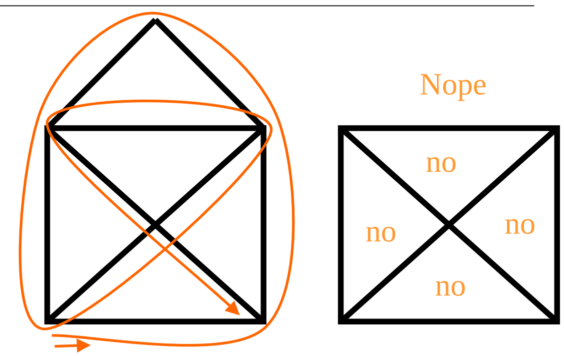




Can you draw these without lifting your pencil, drawing each line only once?
Can you start and end at the same point?

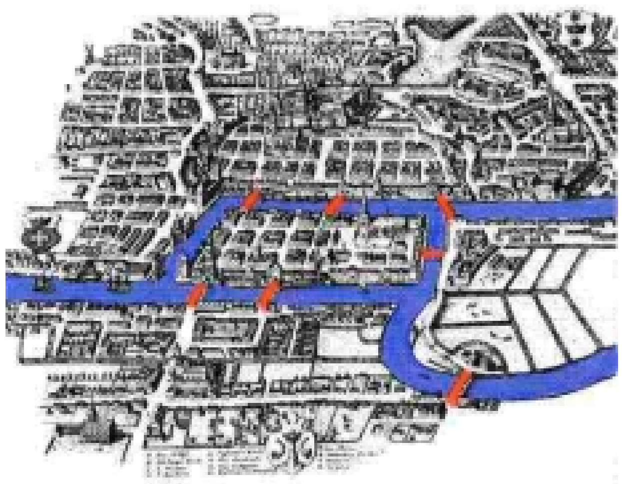
Maybe yes, maybe no





How can we decide without going crazy trying to find one?

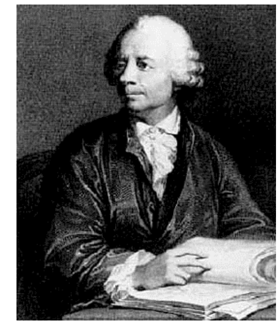
Is it possible to arrange a walking tour which crosses each of the seven bridges exactly once?



The Seven Bridges of Königsberg over the River Pregel in the early 1700's

http://www-gap.dcs.st-and.ac.uk/~history/Miscellaneous/Konigsberg.html

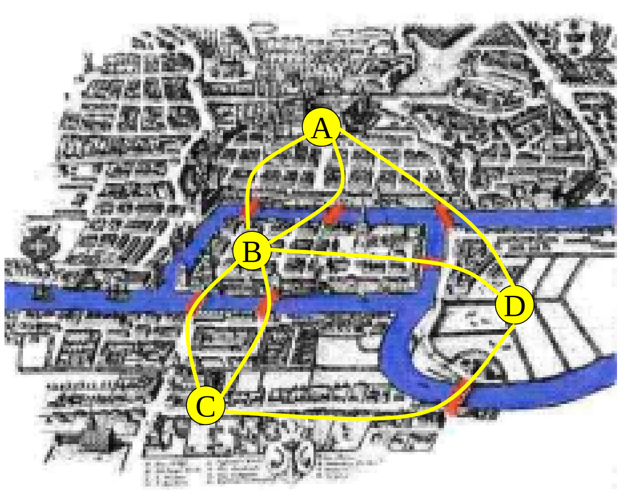
Leonhard Euler (1707-1783)

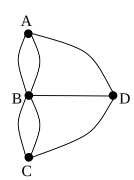


- In 1736 the prolific Leonhard Euler published a solution to the Königsberg bridge problem
  - › *Solutio problematis ad geometriam situs pertinentis*
  - › *The solution of a problem relating to the geometry of position*
- Considered to be an important founding step in the development of graph theory and topology
  - › geometry without measurement

http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Euler.html

Consider this as a graph problem.





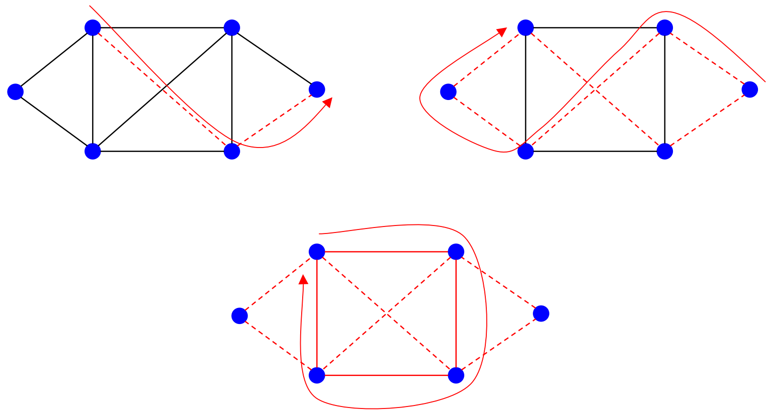
Find a path that traverses every edge exactly once

Is it possible to arrange a walking tour which crosses each of the seven bridges exactly once?

Euler paths and circuits

- An Euler circuit in a graph G is a circuit containing every edge of G once and only once
  - › circuit - starts and ends at the same vertex
- An Euler path is a path that contains every edge of G once and only once
  - › may or may not be a circuit

An Euler Circuit



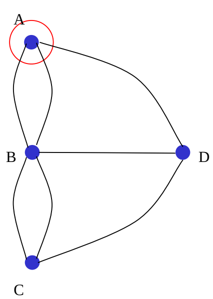
When?

- A connected graph has an Euler circuit if and only if *each of its vertices is of even degree*
  - › At every vertex, need one edge to get in and one edge to get out (or one to get out and one to get back in)
- A connected graph has an Euler path but not an Euler circuit if and only if *it has exactly two vertices of odd degree*
  - › the first and last vertices are distinct
  - › remember that an Euler circuit is also an Euler path

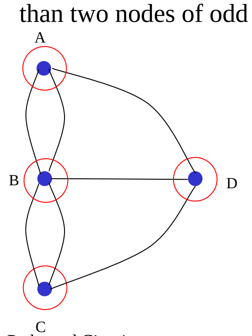
Is it possible to arrange a walking tour which crosses each of the seven bridges exactly once?

Can you find a path that traverses every edge exactly once?

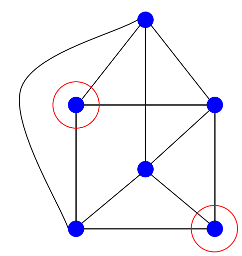
Euler Circuit? No, not all nodes are of even degree.



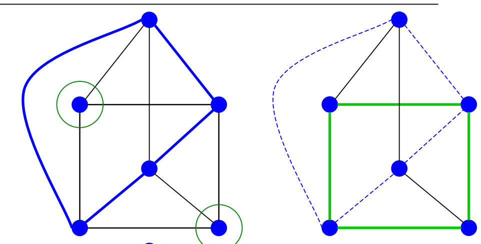
Euler Path? No, there are more than two nodes of odd degree.

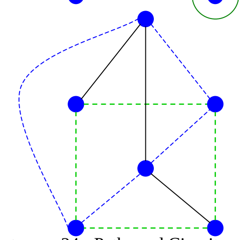


Euler Circuit or Path or None?



Euler Circuit? No, not all nodes are of even degree.



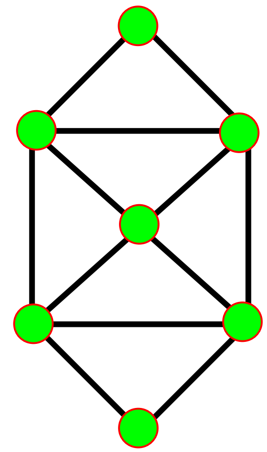


Euler Path? Yes, exactly two nodes of odd degree.

Euler Circuit Problem

- Problem: Given an undirected graph G = (*V*,*E*), find an Euler circuit in G
- Can check if one exists in linear time
  - › check degree of each vertex for the patterns previously described
- Given that an Euler circuit exists, how do we *construct* an Euler circuit for G?

Finding an Euler Circuit

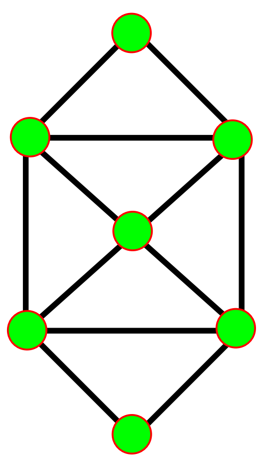


Line segments = edges
Junctions = vertices

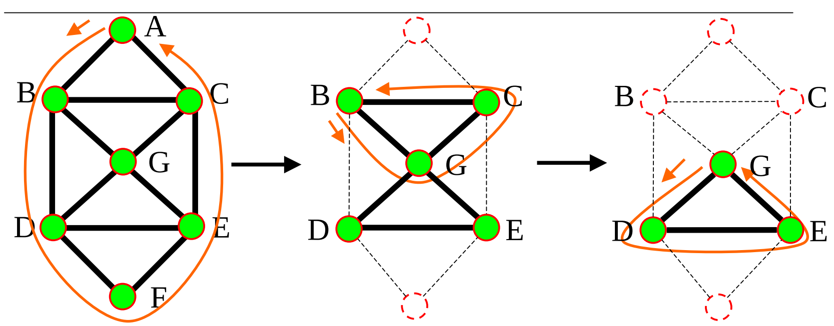
Can you traverse all edges exactly once, starting and finishing at the same vertex?

Depth First Search and then Splice

- Basic Euler Circuit Algorithm:
  - › Do a depth-first search (DFS) from a vertex until you are back at this vertex
  - › Pick a vertex on this path with an unused edge and repeat 1.
  - › Splice all these paths into an Euler circuit
- Running time = O(|*V*| + |*E*|)



Euler Circuit Example



DFS(A) :
A B D F E C A

DFS(B) :
B G C B

Splice at G

DFS(G) :
G D E G

Splice at B A B G C B D F E C A

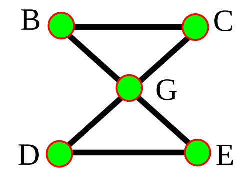
A B G D E G C B D F E C A

Hamiltonian Circuits

- Euler circuit
  - › A cycle that goes through each *edge* exactly once
- Hamiltonian circuit
  - › A cycle that goes through each *vertex* exactly once
- They sound very similar, but they aren't at all
- The algorithms to analyze these circuits are at opposite ends of the complexity spectrum

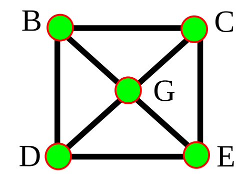
Hamiltonian Circuit Examples

- Does graph **I** have:
  - › An Euler circuit?
  - › A Hamiltonian circuit?



I

- Does graph **II** have:
  - › An Euler circuit?
  - › A Hamiltonian circuit?



II

Finding Hamiltonian Circuits

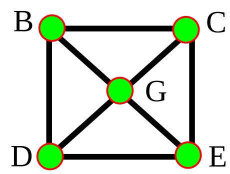
- Problem: Find a Hamiltonian circuit in a graph G = (*V*,*E*)
  - › Sub-problem: Does G contain a Hamiltonian circuit?
  - › Is there an easy (linear time) algorithm for checking this?

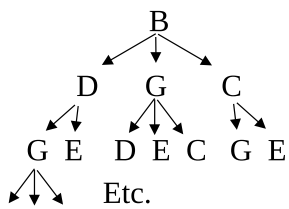
Finding Hamiltonian Circuits

- Does G contain a Hamiltonian circuit?
  - › No known easy algorithm for checking this…
- Try this
  - › Search through *all paths* to find one that visits each vertex exactly once
  - › Can use your favorite graph search algorithm (DFS!) to find various paths
  - › This is an *exhaustive search* ("brute force") algorithm

Exhaustive Search Algorithm Analysis

- How many paths?
- Can depict these paths as a *search tree*
- Let the average branching factor of each node in this tree be B (= average size of adjacency list for a vertex)
- |*V*| vertices, each with ≈ B branches
- Total number of paths ≈ B·B·B … ·B = $O(B^{|V|})$
- Worst case → Exponential time!





Search tree of paths from B

How bad is exponential time?

N	log N	N log N	N^2	2^N
1	0	0	1	2
2	1	2	4	4
4	2	8	16	16
10	3	30	100	1024
100	7	700	10,000	1,000,000,000,000,00, 000,000,000,000,000
1000	10	10,000	1,000,000	Fo'gettaboutit!
1,000,000	20	20,000,000	1,000,000,000,000	ditto
1,000,000,000	30	30,000,000,000	1,000,000,000,000,000,000	mega ditto plus

Polynomial vs Exponential Time

- Most of our algorithms have been O(log N), O(N), O(N log N) or $O(N^2)$ running time for inputs of size N
  - These are all *polynomial time* algorithms
  - Their running time is $O(N^k)$ for some k > 0
- Exponential time B^N is asymptotically worse than *any* polynomial function N^k for any k
  - For any k, N^k is $o(B^N)$ for any constant B > 1
- Polynomial time algorithms are "fast" algorithms
- Exponential time algorithms are "not fast"
  - or "dog slow" to use the technical term

The complexity class P

- The set P is defined as the set of all problems that can be *solved* in polynomial worse case time
  - this is the polynomial time complexity class
  - contains problems whose time complexity to solve is $O(N^k)$ for some k
- Examples of problems in P
  - searching, sorting, topological sort, single-source shortest path, Euler circuit, etc.

The complexity class NP

- The set NP is the set of all problems for which a given *candidate solution* can be *checked* in polynomial time
- Example of a problem in NP:
  - › Hamiltonian circuit problem
  - › Given a candidate path, can test in linear time if it is a Hamiltonian circuit – just check if all vertices are visited exactly once in the candidate path, repeating only the start/finish vertex

Nondeterministic Polynomial time

- Why “nondeterministic”?
  - › A nondeterministic algorithm is free to correctly choose the next step to execute on the path to a solution
  - › Corresponds to algorithms that can search all possible solutions in parallel and pick the correct one
- If we can do this in polynomial time, then we can check a solution in polynomial time
- Nondeterministic algorithms don’t exist – purely theoretical idea invented to understand how hard a problem could be