Circuits

CSE 373

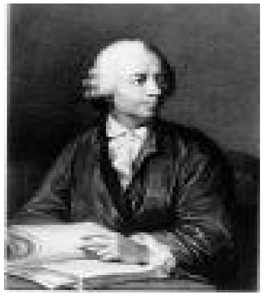
Data Structures

Readings

- Reading
  - Alas not in your book. So it won't be on the final!

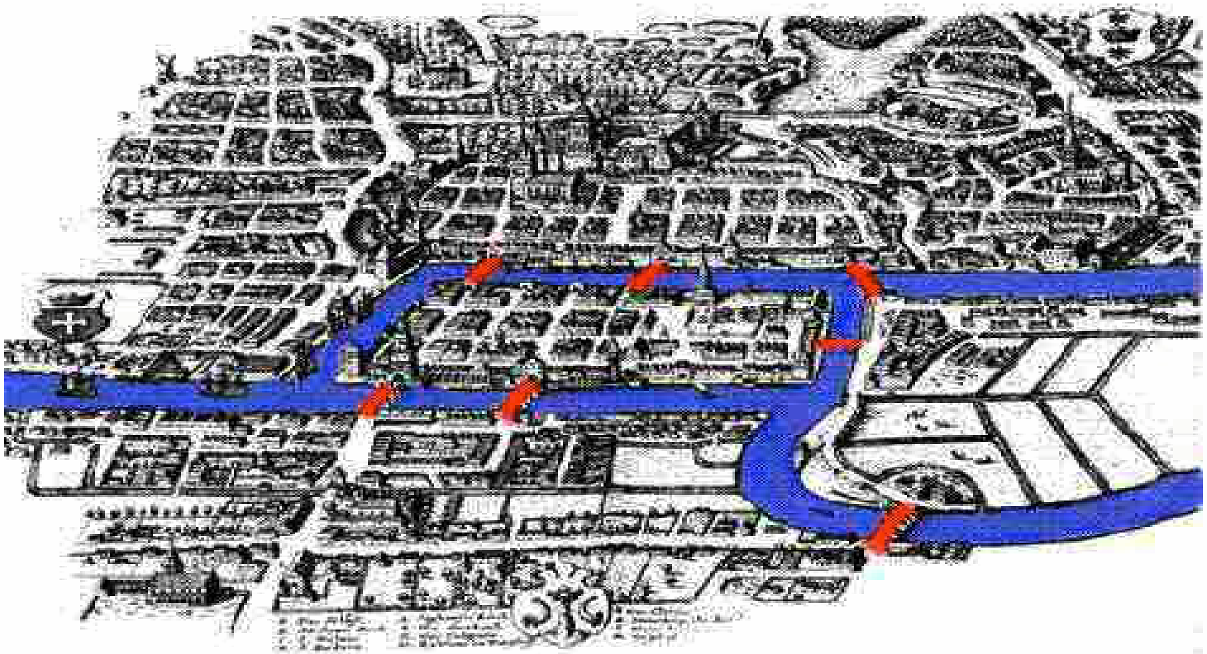
Euler

- Euler (1707-1783): might be the most prolific mathematician of all times (analysis, differential geometry, number theory etc.)
  - For example, *e,* the natural base for logarithms, is named after him; he introduced the notation f(x), the notation for complex numbers (a + *i* b) ….
  - Contributions in optics, mechanics, magnetism, electricity
  - “Read Euler, read Euler, he is our master in everything” (Quote from Laplace a 19th Century French mathematician)



Euler and Graph Theory

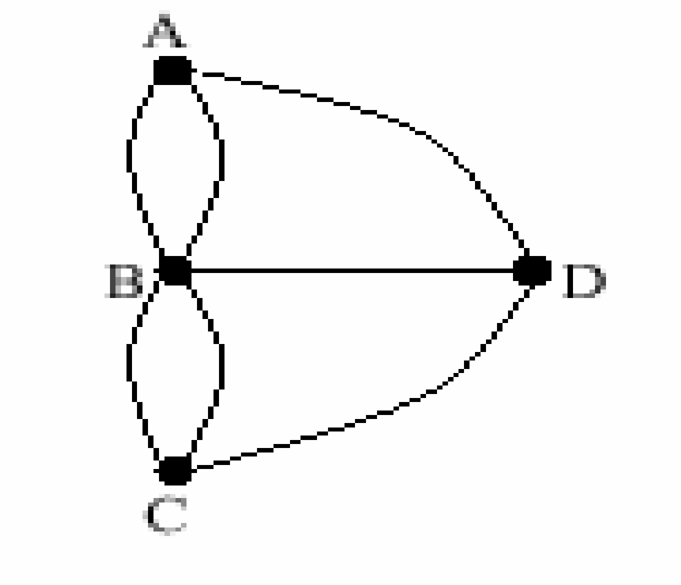
Is it possible to arrange a walking tour which crosses each of the seven bridges exactly once?



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

The Seven Bridges Problem

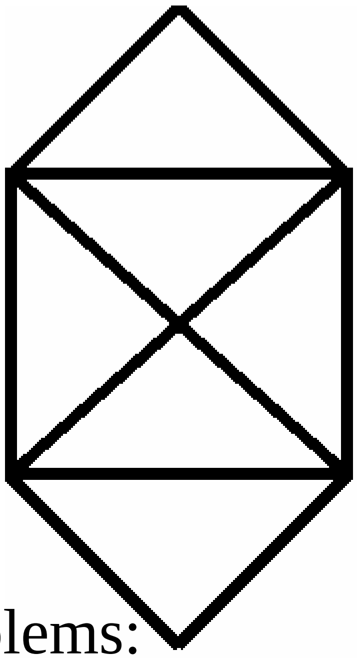
- Each “area” is a vertex
- Each bridge is an edge



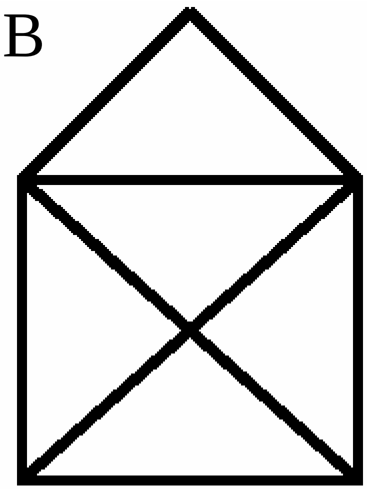
Find a path that traverses each edge exactly once

Related Problems: Puzzles

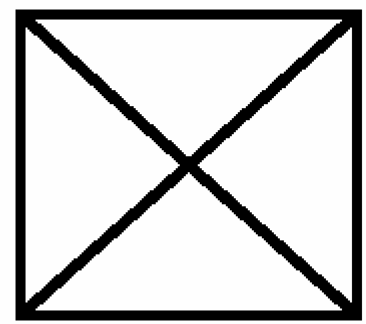
A



B



C



Two problems:

1) Can you draw these without lifting your pen, drawing each line only once
2) Can you start and end at the same point.

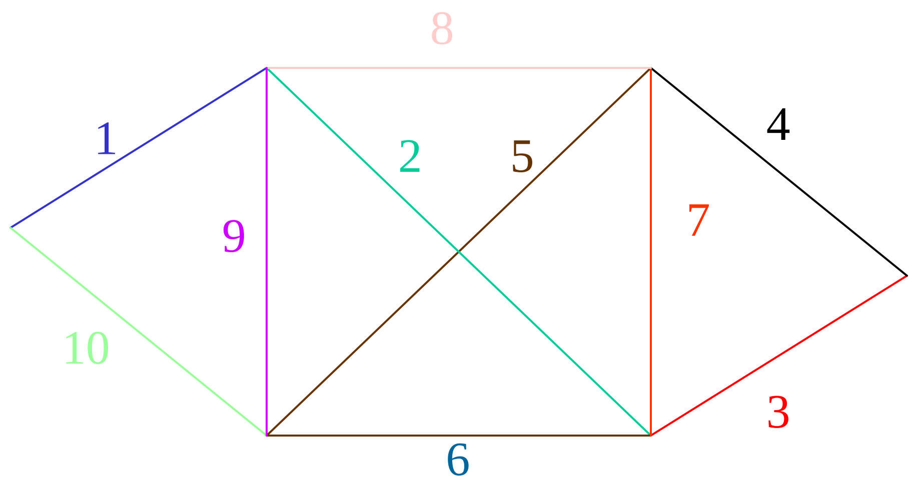
Related Problems: Puzzles

- Puzzle A: 1) yes and 2) yes
- Puzzle B: 1) yes if you start at lowest right (or left) corner and 2) no
- Puzzle C: 1) no and 2) no

Euler Paths and Circuits

- Given G(V,E), an Euler path is a path that contains each edge once (Problem 1)
- Given G(V,E) an Euler circuit is an Euler path that starts and ends at the same vertex (Problem 2)

An Euler Circuit for Puzzle A



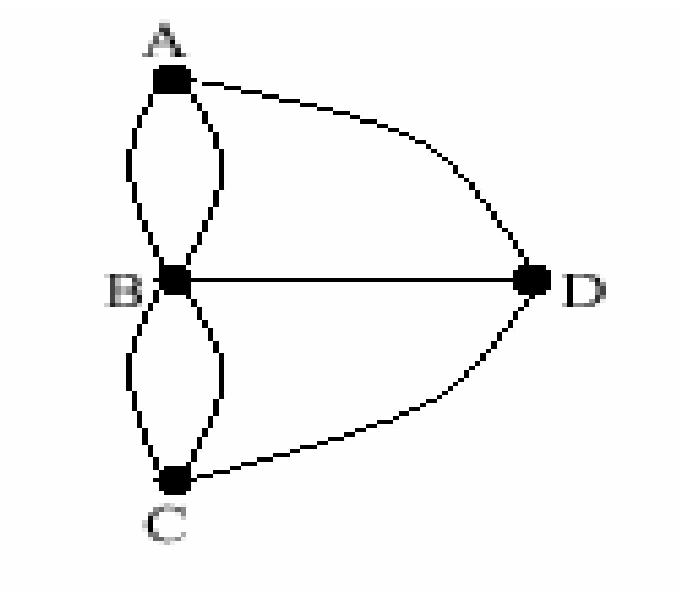
Euler Circuit Property

- A graph has an Euler circuit if and only if it is connected and all its vertices have even degrees (cf. Puzzle A)
  - › Necessary condition (only if): a vertex has to be entered and left, hence need an even number of edges at each vertex
  - › Sufficient condition: by construction (linear time algorithm)

Euler Path Property

- A graph has an Euler path if and only if it is connected and exactly two of its vertices have odd degrees (cf. Puzzle B)
  - › One of the vertices will be the start point and the other one will be the end point
    - to construct it, add an edge (start,end). Now all vertices have even degrees. Build the Euler circuit starting from “start” and at the end delete the edge (start,end).

Back to Euler Seven Bridges



Sorry, no Euler Circuit (not all vertices have even degrees)

Sorry, no Euler path (more than 2 vertices have odd degrees)

Finding an Euler Circuit

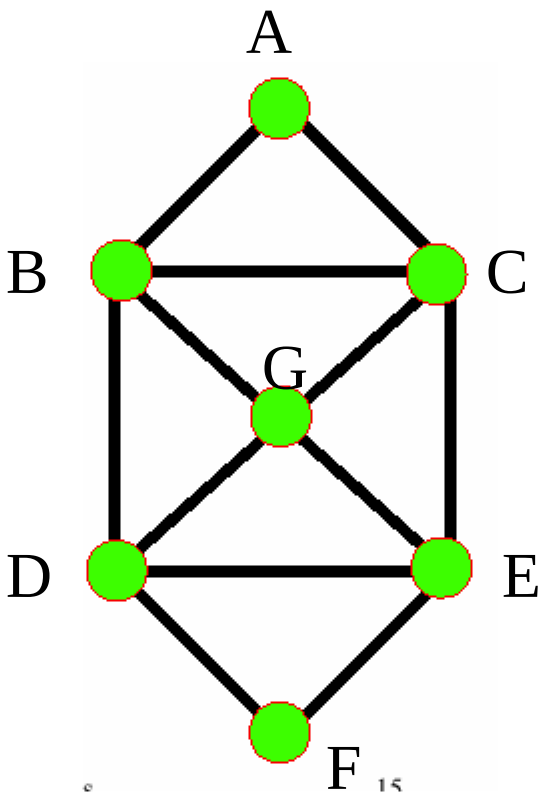
- Check than one exists (all vertices have even degrees)
- Starting from a vertex v_i (at random)

  While all edges have not been "marked"

  DFS(v_i) using unmarked edges (and marking them) until back to v_i

  Pick a vertex v_j on the (partial) circuit with a remaining unmarked edge and repeat the loop; Splice this new circuit in the old one and repeat.

Example

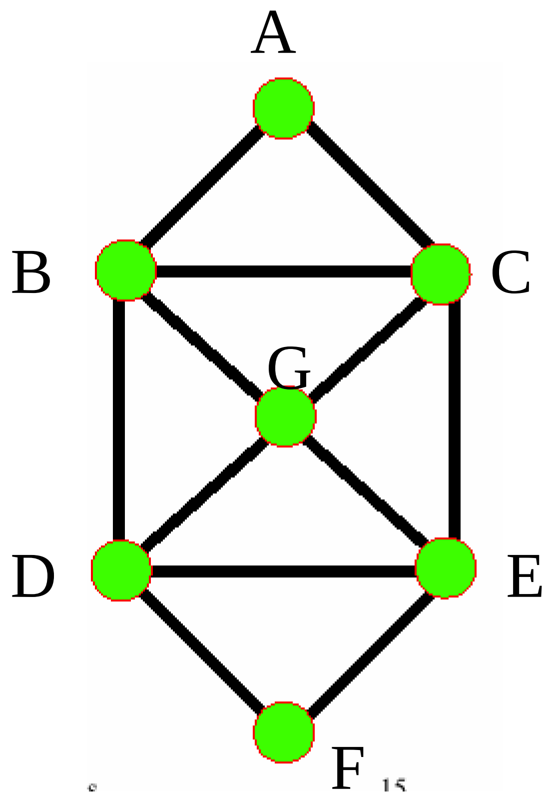


Pick vertex A

DFS(A) yields circuit ABCA and edges (A,B),(B,C) and (C,A) are marked

Pick a vertex in circuit with an unmarked edge, say B

Example (ct'd)



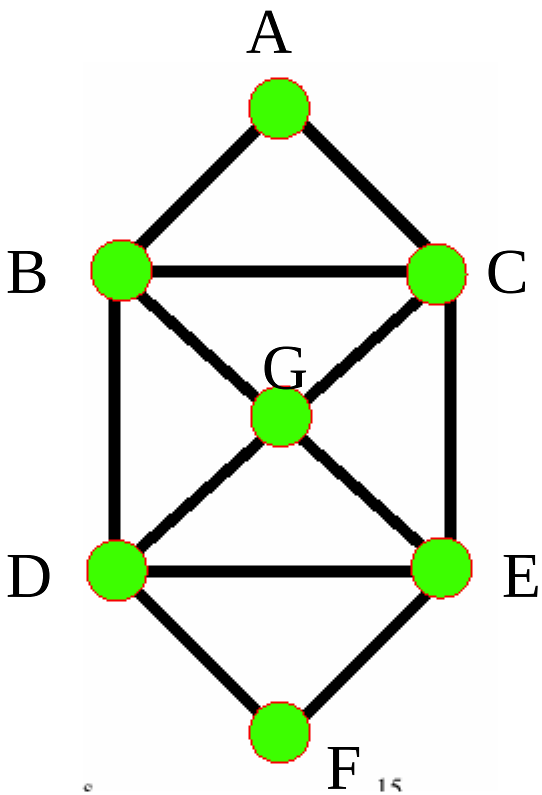
ABCA

Picking B yields circuit BDECGB (note that when we reached C, we had to go to G since (C,B), (C,A) and (C,E) were marked

Slice the green circuit in the blue one

ABDECGBCA

Example (end)



ABDECGBCA

Pick vertex with unmarked edge D

DFS(D) yields DFEGD

Splicing yields Euler circuit

ABDFEGDECGBCA

Euler Circuit Complexity

- Find degrees of vertices: O(m)
- Mark each edge once (in each direction) thus traverse each edge once: O(m)
  - This might require slightly improved adjacency list
- Splice the circuits:at most n cycles (use linked lists: O(n))
- Linear time O(n+m)

Hamiltonian Circuit

- A Graph G(V,E) has an hamiltonian circuit if there exists a path which goes through each vertex exactly once
- Seems closely related to Euler circuit
- It is NOT!
- Euler circuit can be solved in linear time
- Hamiltonian circuit requires exhaustive search (exponential time) in the worse case

Sir William Hamilton

- Irish mathematician (1805-1865)

Examples

Does Graph I have

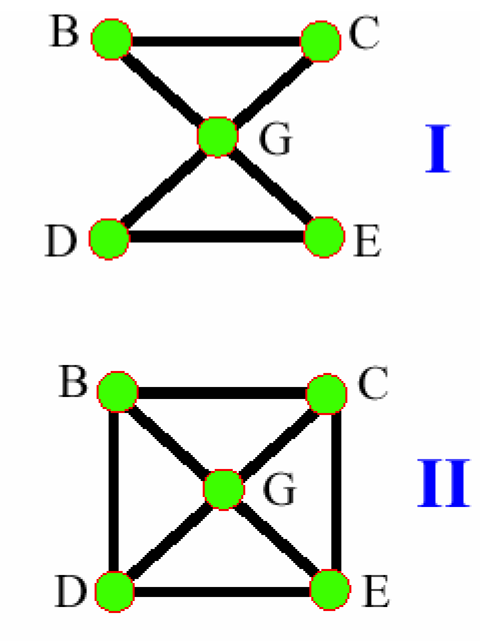
an Euler circuit?

an Hamiltonian circuit?

Does Graph II have

an Euler circuit?

an Hamiltonian circuit?



Finding Hamiltonian Circuits

- Apparently easier “Yes” or “No” question: “Does G contain an Hamiltonian circuit?”
  - › NO known “easy” algorithm, i.e., no algorithm that runs in $O(n^p)$, or polynomial time in the number of vertices
  - › Requires exhaustive search (brute force)

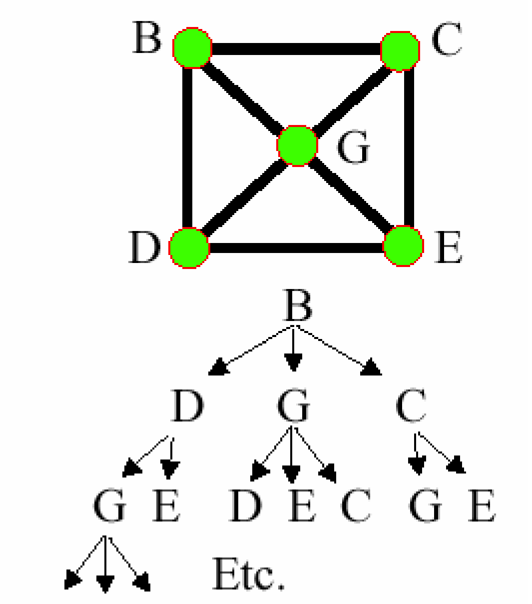
Example of Exhaustive Search

How many paths?

Let B be the average branching factor at each node for DFS

Total number of paths to be examined B.B.B….B = $O(B^n)$

Exponential time!



Search tree of paths from B

Time Complexity of Algorithms

- If one has to process n elements, can't have algorithms running (worse case) in less than O(n)
- But what about binary search, deletemin in a heap, insert in an AVL tree etc.?
  - › The input has been preprocessed
  - › Array sorted for binary search, buildheap for the heap, AVL tree already has AVL property etc. all ops that took at least O(n)

The Complexity Class P

- Most of the algorithms we have seen have been polynomial time $O(n^p)$
  - › Searching (build the search structure and search it), sorting, many graph algorithms such as topological sort, shortest-path, Euler circuit
  - › They form the class P of algorithms that can be solved (worse case) in polynomial time

Are There Problems not in P?

- For some problems, there are no known algorithms that run (worse case) in polynomial time
  - › Hamiltonian circuit
  - › Circuit satisfiability
    - Given a Boolean formula find an assignment of variables such that the formula is true (even if only 3 variables)
  - › Traveling Salesman problem
    - Given a complete weighted graph G and an integer K, is there a circuit that visits all vertices only once of cost less than K
  - › Etc.

Undecidability

- There are problems that cannot be solved algorithmically: they are undecidable
- The most well-known one is the Turing Halting Problem
  - Turing proved that it is impossible to write a "computer program" that can read another computer program, and, if that program will run forever without stopping, tell us, after some finite (but unbounded) time this fact.