Algorithms on Graph

Xiaofeng Gao

Department of Computer Science and Engineering
Shanghai Jiao Tong University, P.R.China

Algorithm Course @ Shanghai Jiao Tong University

Outline

Königsberg

Once upon a time there was a city called Königsberg in Prussia.

It is the capital of **Kingdom of Prussia** until 1945.

The literal meaning of Königsberg is "King's Mountain".

Centre of learning for centuries, being home to *Christian Goldbach*, *David Hilbert*, *Immanuel Kant* ...

Position of Konigsberg

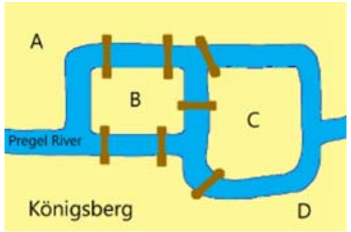
Seven Bridge

Pregel River is passing through Königsberg.

It separated the city into two mainland area and two islands.

There are seven bridges connecting each area.

A Tour Question: Can we wander around the city, crossing each bridge once and only once?



Euler's Solution

Leonhard Euler Solved this problem in 1736.

Published the paper "The Seven Bridges of Königsberg".

The first negative solution laid the foundations of Graph Theory and pre-figured the idea of topology.

The Seven Bridges of Königsberg

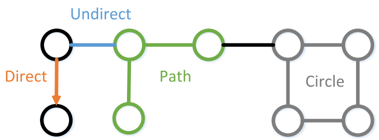
1.

The branch of geometry that deals with magnitudes has been zealously studied throughout the past, but there is another branch that has been almost unknown up to now; Leibniz spoke of it first, calling it the "geometry of position" (*geometria situs*). This branch of geometry deals with relations dependent on position alone, and investigates the properties of position; it does not take magnitudes into consideration, nor does it involve calculation with quantities. But as yet no satisfactory definition has been given of the problems that belong to this geometry of position or of the method to be used in solving them. Recently there was announced a problem that, while it certainly seemed to belong to geometry, was nevertheless so designed that it did not call for the determination of a magnitude, nor could it be solved by quantitative calculation; consequently I did not hesitate to assign it to the geometry of position, especially since the solution required only the consideration of position, calculation being of no use. In this paper I shall give an account of the method that I discovered for solving this type of problem, which may serve as an example of the geometry of position.

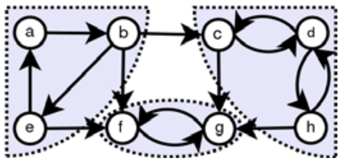
2. The problem, which I understand is quite well known, is stated as follows: In the town of Königsberg in Prussia there is an island A, called

Definition

- Vertex, (Direct/Undirect) Edge, Path, Circle



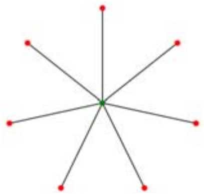
- Undirected Graph: $G = (V, E)$, V: vertex, E: edges
- Directed Graph: $G = (V, A)$, V: vertex, A: arcs (directed edges)
- Strongly Connected Component (Every vertex is reachable from every other vertex)



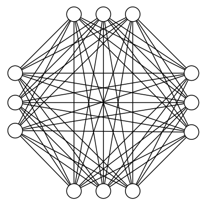
Well-Known Results

- Complete Graph K_n
- Bipartite Graph $K_{m,n}$
- Star $K_{1,n}$
- r-Partite Graph $K_{r(m)}$
- Subgraph $H \subset G$: Spanning/Induced subgraph
- Handshaking Theorem:

$$\sum_{v \in V} d(v) = 2|E|$$



$K_{1,7}$



$K_{4(3)}$

Algorithms on Graphs

Graph Decomposition

- Depth-First Search $\rightarrow$ Topological Sort, DAG, Stack
- Breadth-First Search $\rightarrow$ Cardinality Shortest Path, Queue
- Minimum Spanning Tree $\rightarrow$ Prim, Kruskal, Circle-Delete

Shortest Path

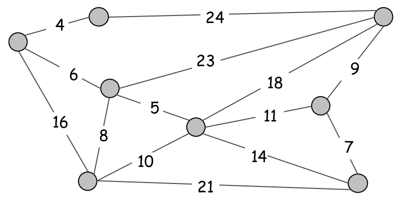
- Single-Source Shortest Path $\rightarrow$ Dijkstra, Bellman-Ford
- All-Pairs Shortest Path $\rightarrow$ Matrix, Floyd-Warshall, Johnson's

Maximum Flow

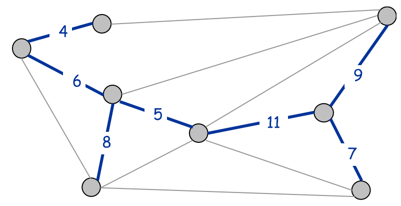
- Max-Flow Min-Cut Theorem
- Ford-Fulkerson Algorithm
- Edmond-Karp Enhancement (Augmenting Path)

Definition of Minimum Spanning Tree

Given a connected graph $G = (V, E)$ with real-valued edge weight C_e, a Minimum Spanning Tree (MST) is a subset of the edges $T \subseteq E$ such that T is a spanning tree whose sum of edge weights is minimized.



$G = (V, E)$



$T, \ \Sigma_{e \in T} \, c_e = 50$

Algorithms of Minimum Spanning Tree

Classical Algorithms

- Prim: maintain an optimal subtree
- Kruskal: maintain min-weight acyclic edge set
- Reverse-Delete: circle-deletion
- Borüvka Algorithm

Fundamental Results

- All greedy approach with exchange property
- Correctness proof: cycle/cut property
- Efficiency: time complexity $\rightarrow$ heap