Data Structures and Algorithms 2
Design and Analysis of Algorithms

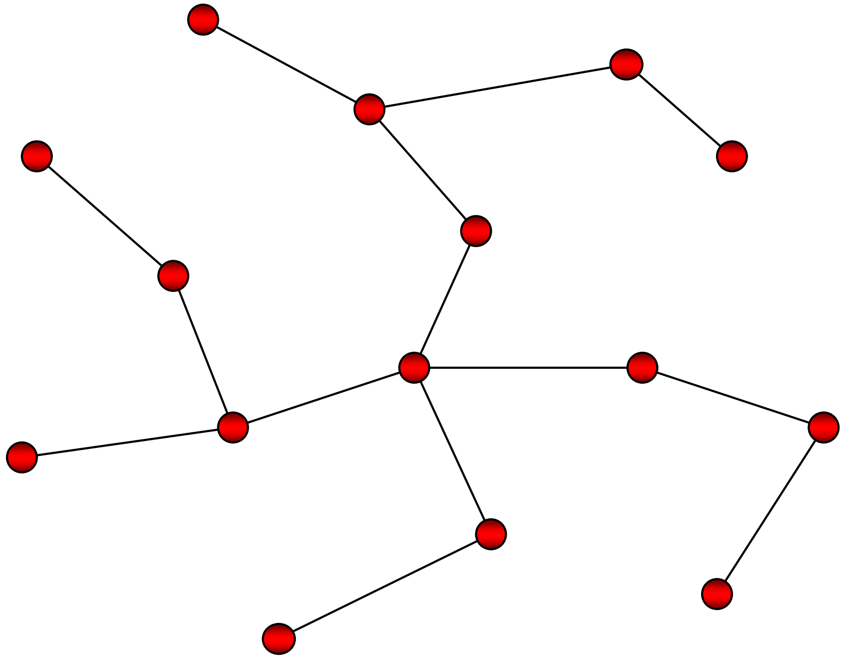
Oswin Aichholzer

Birgit Vogtenhuber

Institute of Software Technology

Example 1: Minimum Spanning Tree

Connect n points (cities) with a network of minimal length

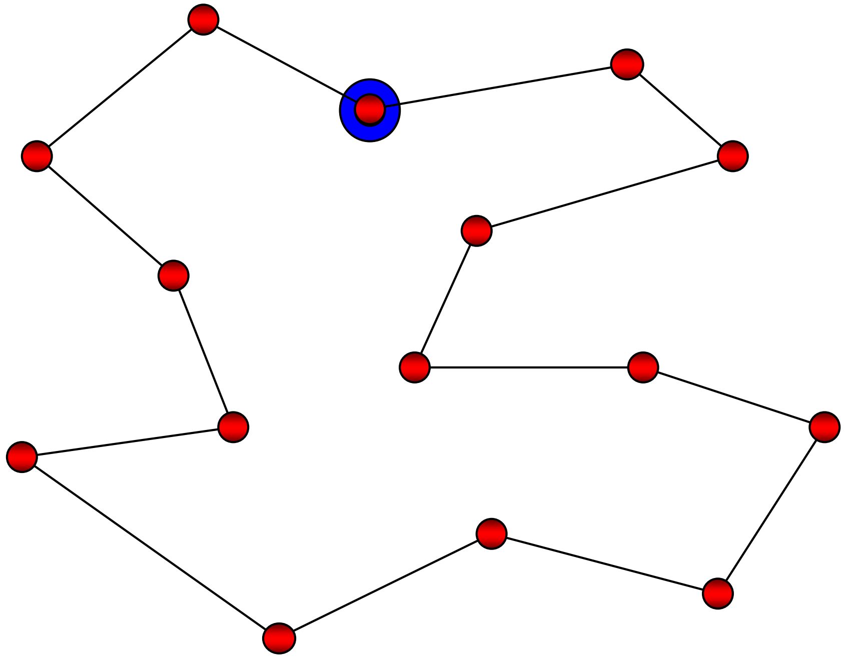


- Formal:
  - Given n points in the plane
  - Compute a network which connects all points and has minimal total length.
- Applications:
  - Networks of streets, public transport, phone lines, computer connections, water pipes, …
- Optimization Problem

How fast can you compute the minimum spanning tree for a set of n points?

Example 2: Traveling Salesperson

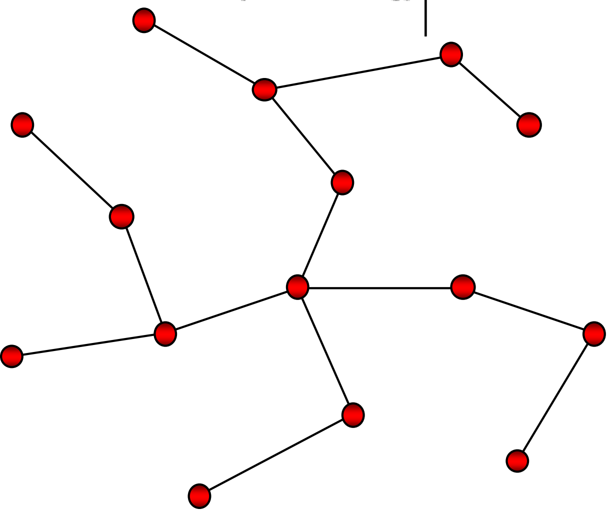
Find a round trip through n cities (points) with minimal length



- Formal:
  - Given n points in the plane
  - Compute a cycle which visits each points precisely once and is as short as possible.
- Application:
  - Parcel delivery
  - Drilling of holes in printed circuit boards
- Known as traveling salesperson (TSP)
- Optimierungsproblem

How fast can you compute the shortest tour through n points in the plane?

Comparing the two Problems

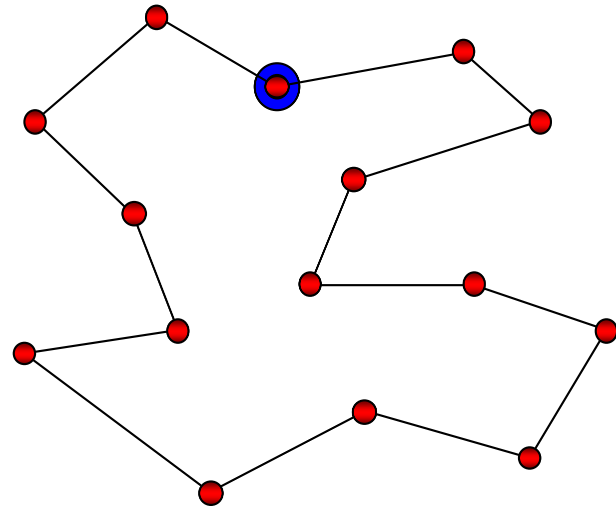


ST **vs.** **TSP**

Given n points

Compute a graph that connects all points
with minimum total length

arbitrary **closed cycle**



Running time for good deterministic algorithms:

ST		TSP	
n = 10:	0.01 seconds	n = 10:	0.01 seconds
n = 20:	0.025 seconds	n = 20:	10.2 seconds
n = 30:	0.04 seconds	n = 30:	2 h 55 min
n = 100:	0.2 seconds	n = 100:	

Graz University of Technology

4000000000000000000 years? Really?

- The Traveling salesperson problem is an NP-complete problem.
  - There are thousands of problems which belong to that class.
  - There does not exist any efficient deterministic algorithm for these problems, meaning that all known approaches need exponential running time.
  - Algorithms with exponential running time are not practical.
  - We do NOT have a proof that efficient algorithms can not exist.
  - If you can solve one NP-hard problem efficiently, then you can solve all of them!
  - To determine the actual complexity of this class is one of the most prominent open problems in computer science, known under the acronym “P=NP?”.

Graz University of Technology

4000000000000000000 years? And now?

- What can we do if we anyhow want to solve such a problem? For example if we have to produce a printed circuit board with 1500 holes or deliver 50 parcels?
  - In one year we could solve the problem for approx. 41-42 points on a standard PC.
  - Does better hardware help?
  - Assume every particle in the known universe (approx. 10^{79}) would be a super-computer with a computation power of 1000-times of a regular PC. If all would be running since the beginning of the existance of the universe (approx. 13-16 billion years), the they could have solved the problem for n=347 points.
  - Today smart heuristics and approximation methods can solve pratical instances of the traveling salesperson problem for several thousand points easily.
    → **Mathematics and theory of algorithms is more important than better hardware.**
- In contrast: a minimum spanning tree can be computed in near linear time (→ $O(n \log n)$), using efficient sorting algorithms and suitable data structures.
  - For n=347 we can compute a minimum spanning tree on a regular PC in 0.8 seconds.

How can Algorithm Courses help?

- You will learn about design prinziples and new data structures
  - Divide & Conquer, dynamic programming, scanline-principle, …
  - Asymptotic notations for simpler analysis of runtime and memory efficiency for increasing input size
- You will see methods to analyse and classify problems like
  - Identify problems that can be solved efficiently versus NP-hard problems
  - For problems where no efficient deterministic solutions exist we will see heuristics, randomized algorithms, approximation techniques …

Graz University of Technology

Summary / Overview

- Design principles
  - Dynamic programming
  - Scanline-Principle
  - Divide & Conquer
- Asymptotic complexity
  - Time- and memory consumption
  - Recursive Equations
- Geometric algorithms
  - Triangulations
  - Intersection of line segments
  - Convex hulls
- Graph algorithms
  - How to store graphs
  - Searchng in graphs
  - Spanning trees
  - Shortest path
- Complexity theory
  - NP-completness
  - Approximation algorithms and heuristics
- Seleced chapters
  - Depending on the term

Questions?

More information at the course teach center