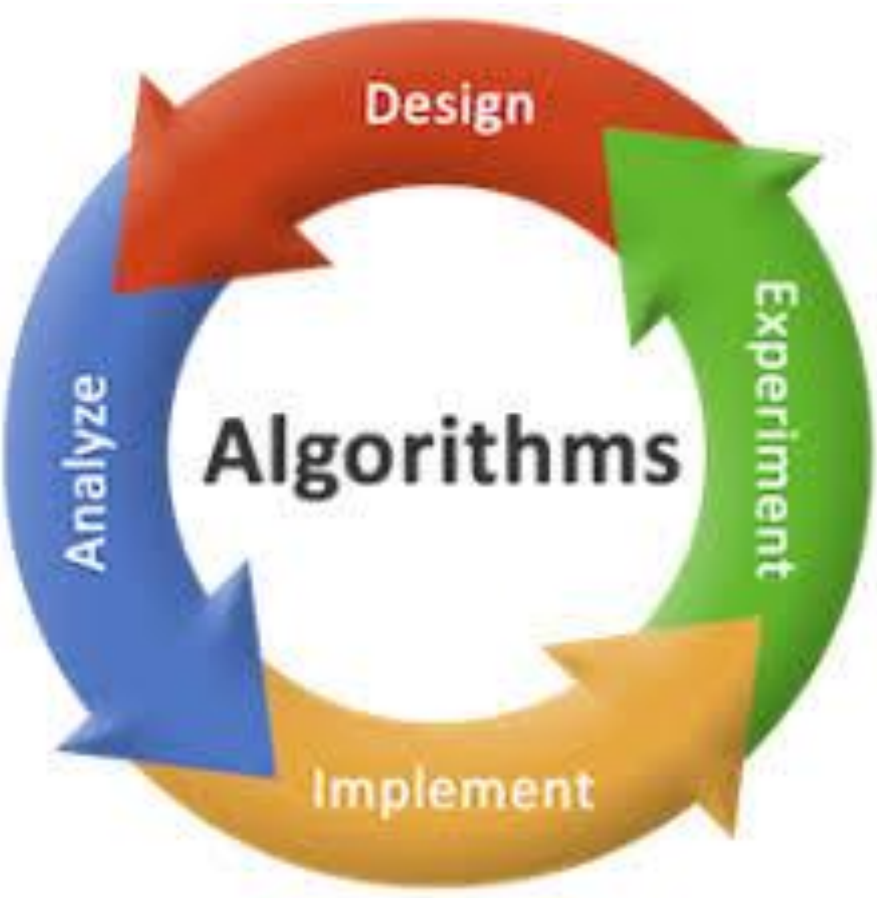


Design & Analysis of Algorithms

Lecture 11

Brute Force-

Travelling Salesman Problem

Travelling Salesman Problem

- The **traveling salesman problem (TSP)** has been an interesting problem to researchers for the last 150 years by
  - its seemingly simple formulation
  - important applications
    - planning, scheduling, logistics and packing
  - interesting connections to other combinatorial problems

Travelling Salesman Problem

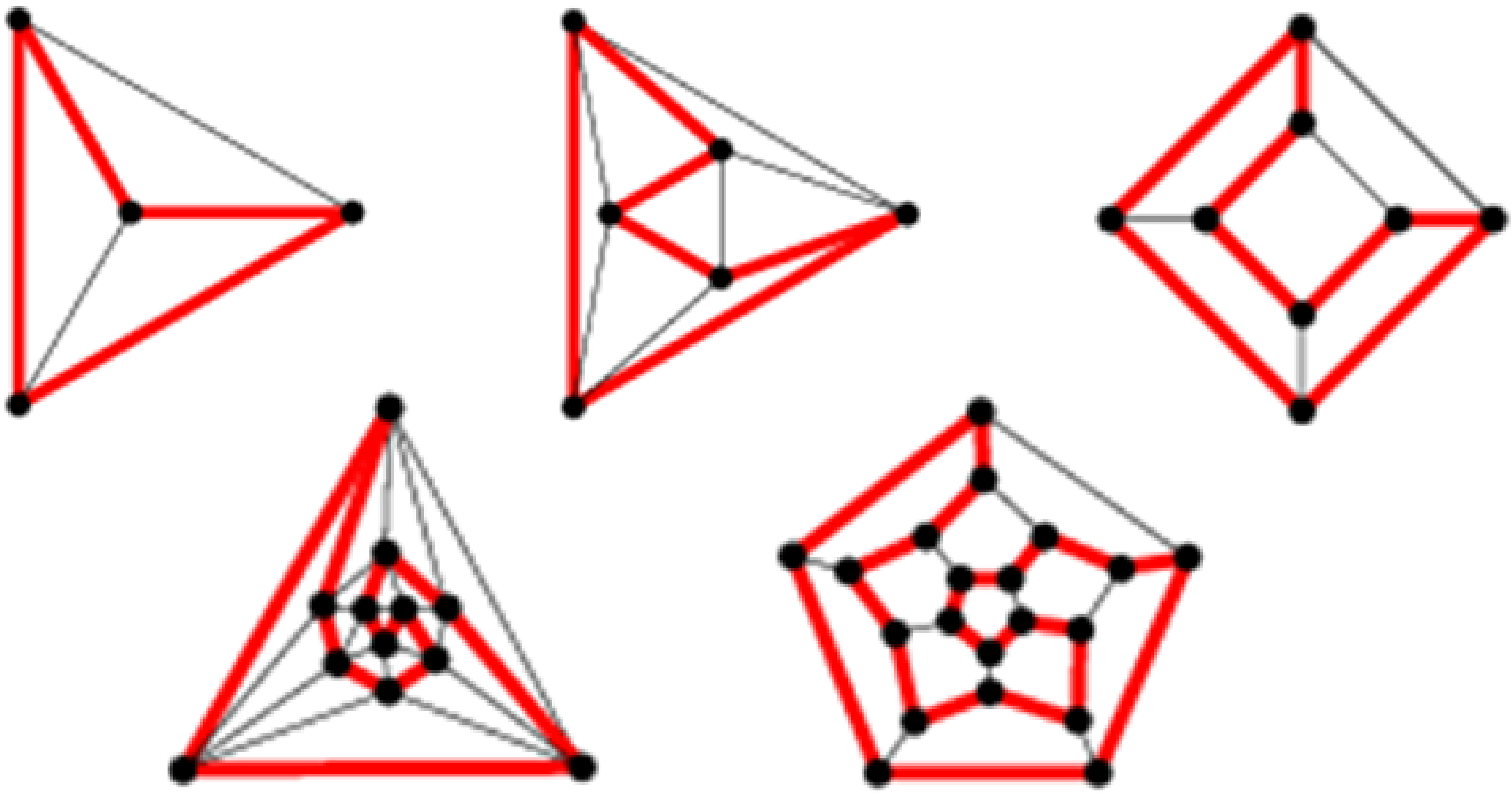
- The problem asks to find the shortest tour through a given set of n cities that visits each city exactly once before returning to the city where it started.
- The problem can be conveniently modeled by a **weighted graph**, with the graph's vertices representing the cities and the edge weights specifying the distances.

Travelling Salesman Problem

- Then the problem can be stated as the problem of finding the shortest ***Hamiltonian circuit/cycle*** of the graph.

- A Hamiltonian circuit/cycle is defined as a cycle that passes through all the vertices of the graph exactly once.

Travelling Salesman Problem



Examples: Hamiltonian circuit/cycle

Travelling Salesman Problem

- A Hamiltonian circuit can also be defined as a sequence of $n + 1$ adjacent vertices

$$v_{i_0}, v_{i_1}, \ldots, v_{i_{n-1}}, v_{i_0},$$

  where the first vertex of the sequence is the same as the last one and all the other $n - 1$ vertices are distinct.

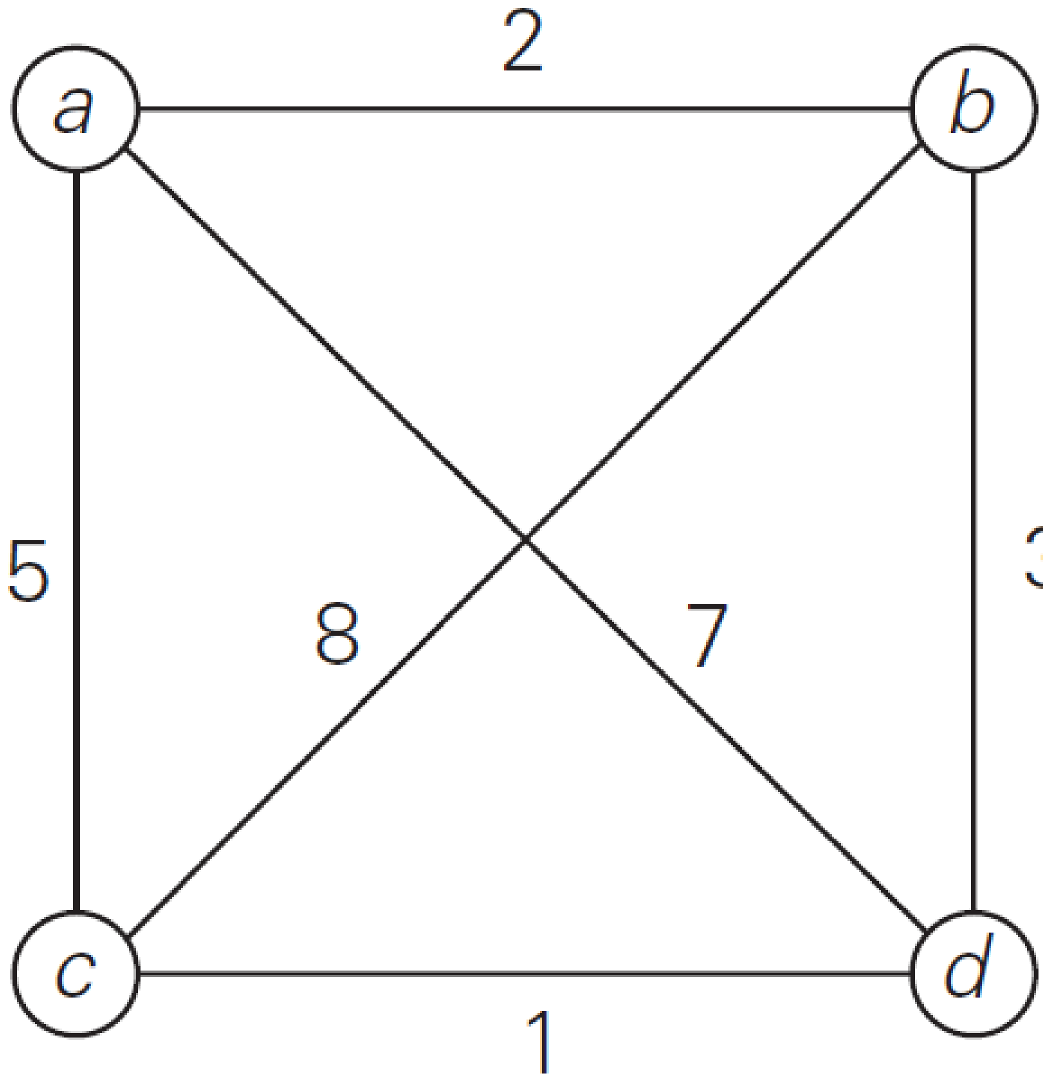
Travelling Salesman Problem

- Further, we can assume, that all circuits start and end at one particular vertex (cycles).

- Thus, we can get all the tours by
  - generating all the permutations of $n - 1$ intermediate cities,
  - compute the tour lengths, and
  - find the shortest among them.

Travelling Salesman Problem

Example:



Travelling Salesman Problem

Solution:

Tour	Length	
$a \dashrightarrow b \dashrightarrow c \dashrightarrow d \dashrightarrow a$	$l = 2 + 8 + 1 + 7 = 18$	
$a \dashrightarrow b \dashrightarrow d \dashrightarrow c \dashrightarrow a$	$l = 2 + 3 + 1 + 5 = 11$	optimal
$a \dashrightarrow c \dashrightarrow b \dashrightarrow d \dashrightarrow a$	$l = 5 + 8 + 3 + 7 = 23$	
$a \dashrightarrow c \dashrightarrow d \dashrightarrow b \dashrightarrow a$	$l = 5 + 1 + 3 + 2 = 11$	optimal
$a \dashrightarrow d \dashrightarrow b \dashrightarrow c \dashrightarrow a$	$l = 7 + 3 + 8 + 5 = 23$	
$a \dashrightarrow d \dashrightarrow c \dashrightarrow b \dashrightarrow a$	$l = 7 + 1 + 8 + 2 = 18$	

Travelling Salesman Problem

- Three pairs of tours that differ only by their direction.
- Hence, we could **cut** the number of vertex permutations by **half**.
- We could, for example, choose any two intermediate vertices, say, *b* and *c*, and then consider only permutations in which *b* precedes *c*. (This trick implicitly defines a tour's direction.)

Travelling Salesman Problem

- The total number of permutations needed is still

$$\frac{1}{2}(n-1)!,$$

which makes the exhaustive-search approach impractical for all but very small values of *n*.

References

- **Chapter 3**: Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Pearson Education, Third Edition, 2017.

Homework

- Cryptarithmetic/Alphametic Problems
  - SEND + MORE = MONEY
  - EAT + THAT = APPLE
  - DONALD + GERALD = ROBERT
- Password Guessing
  - If the password (of size **n**) contains {A-Z, a-z, 0-9, special characters}
- String Matching/Sub-string Matching