Week 1 (2) – Important Problem Types

CST370 – Design & Analysis of Algorithms

Dr. Byun

Computer Science

Lecture Objectives

- After completion of this lecture, you will be able to
  - recognize the importance of algorithms in computer science.
  - identify important problem types in the algorithm class such as sorting, searching, graph problems, etc.

Why do we study algorithms?

- You can **learn a set of standard algorithms** in different computing areas.
- You will be able to **design a new algorithm** to solve a new problem and **analyze the efficiency** of your algorithm.
- It is the **cornerstone of computer science**.
  - Computer programming will not exist without algorithms.
  - It is also considered a common troubleshooting strategy.

Google Recruiter's Email

- The following is a part of an email from a Google recruiter to our student who had a job interview with them.
  - *Many problems will be about algorithms.*
  - ***Algorithm Complexity**: You need to know **Big-O**. If you have difficulty analyzing the basic Big O complexity, you will rarely be hired.*
  - ***Sort**: You should know how to sort.*
  - ***Hashtables**: Probably the single most important single data structure known to mankind. ...*
  - ***Tree**: ...*
  - ***Graph**: ...*
  - ***Other data structures**: As many different data structures and algorithms as possible should be studied.*

Chapter 1: Introduction

- 1.1 What is an Algorithm?
- 1.2 Fundamentals of Algorithmic Problem Solving
- **1.3 Important Problem Types**
- 1.4 Fundamental Data Structures

Important Problem Types

- Watch the video first
  - https://youtu.be/UFC_X5ESsfI
- This section briefly introduces the most important problem types in algorithms.
  - Sorting
  - Searching
  - String processing
  - Graph problems
  - Combinatorial problems
  - Geometric problems
  - Numerical problems

Sorting

- Sorting is to arrange items in either ascending order or descending order.
  - Sometimes, we call it non-ascending order or non-descending order.
- Why is the sorting important in CS?
  - Sorted data provides fast searching.
  - There are many applications which require sorting, and it's the basis of other important algorithms.

Sorting – Time Efficiency

- So far, many sorting algorithms have been proposed.
- **O(n*logn)** is considered the fastest one **for comparison-based sorting**.
  - If you don't know the meaning of the notation O(n*logn), that is fine.
  - We will cover the notation O later.

Stable Sorting

- A sorting algorithm is said to be **stable** if the algorithm keeps the relative order of equal elements in its input.
- Watch this video.

  http://y2u.be/CIG4xjwQ0BM

Stable vs. Unstable Sorting

- Stable sorting algorithms
  - Merge sort, insertion sort, bubble sort, etc.
- Unstable sorting algorithms
  - Quick sort, heap sort, selection sort, etc.
- Don’t misunderstand that stable sorting is always better than unstable sorting.
  - You should choose according to your needs.

In-Place Sorting Algorithm

- An algorithm is said to be ***in-place*** if it does not require a lot of extra memory, except for a few memory units.
  - For example, if your algorithm requires an array to hold the input data and a few primitive variables to conduct the sorting, you can say that your algorithm is “in-place”.
  - However, if your algorithm requires additional array(s), your algorithm is not “in-place”.
- In-place sorting
  - Bubble sort, selection sort, insertion sort, heapsort, quicksort, etc.

Searching

- Finding a given value, called a *search key*, in a given set.
  - e.g., sequential search, binary search, hashing, etc.

String Processing

- String examples
  - Text strings, bit strings, gene sequences, etc.
- One important problem
  - String matching problem: searching for a word in a word file.
- Example
  - Word file: I like the C++ programming and Java programming. They are good languages.
  - You will use “Control-F” to find a word like “pro”.

Puzzle: Three mislabeled baskets (1 of 2)

- Suppose that there are three baskets on a desk.
  - One basket contains an apple, another basket contains an orange, and the last basket contains an apple and an orange.
- Now, let's assume that each basket has a label outside like “Apple”, “Orange”, or “Mix”.
  - The problem is that the labels are always mislabeled. For example, if the label says “Mix”, you can be sure that the basket is never actually a blend of an apple and an orange, i.e. it actually contains either an apple or an orange.

Puzzle: Three mislabeled baskets (2 of 2)

- In this puzzle you can choose **only one basket** to see what is in it.
- Which basket should you choose to label all three properly?

Important Note

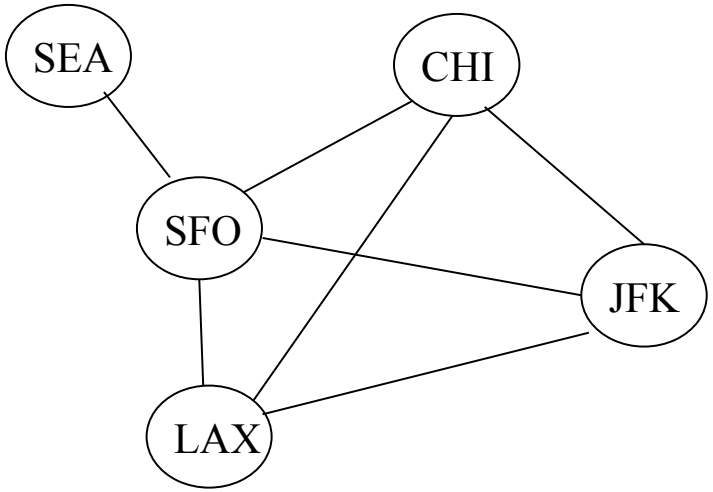
- **Do not see the answer immediately** on the next page.
  - If you do, you will not get any knowledge from this puzzle.

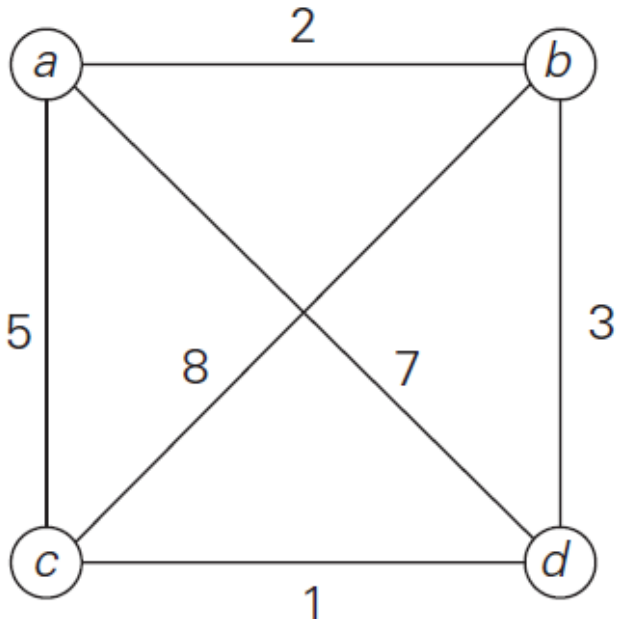
Solution

- This problem can be solved by choosing any basket. It doesn't matter which basket you choose.
  - For instance, let’s say that you select the basket with the label Mix.
  - Since all baskets are mislabeled, the actual content of the box should be either an apple or an orange.
  - If the basket has an apple, you know that the basket with the label Apple should have an orange, and the Orange basket should have the mix of an apple and an orange. Otherwise, all three baskets can’t be mislabeled.
  - Similarly, whatever basket you choose, you can use the information of the selected basket to presume the contents of other boxes.

Graph Problems (1 of 2)

- A graph in computer science is composed of nodes (= vertices) and lines (= edges).
- Example





Graph Problems (2 of 2)

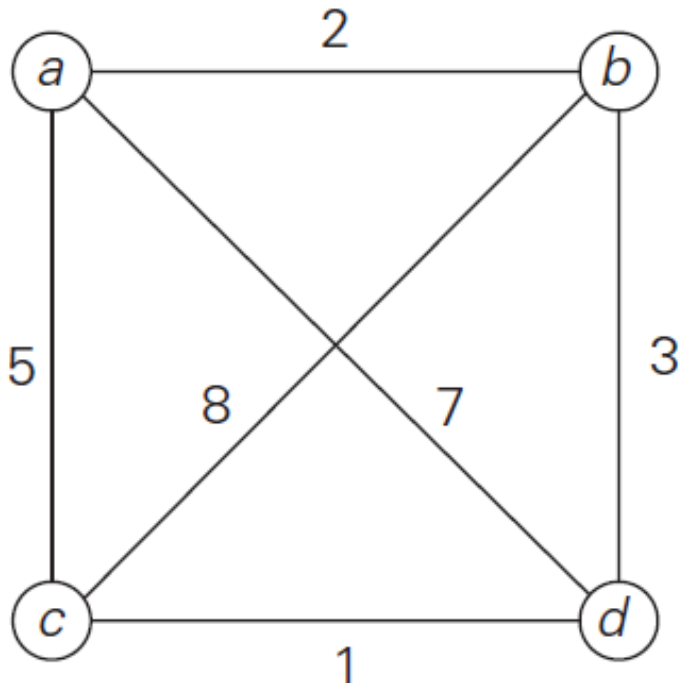
- A graph is used for modeling many real-life applications.
  - e.g.,: transportation, communication networks, project scheduling, etc.
- Graph traversal problem
- Shortest-path problem

Traveling Salesman Problem (TSP)

- Watch the video first
  - https://youtu.be/67DMNFzzIpo
- It is one of the best known graph problems in computer science.
- The travelling salesman problem (TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, **what is the shortest possible route** that **visits each city exactly once** and **returns to the origin city**?"

TSP Example

- Assume that you start a traversal for the following graph from the node *a*.
  - What is the optimal travel path and travel cost?



TSP Example – Solution

- There are two optimal paths (or tours)

  (1) a → b → d → c → a

  (2) a → c → d → b → a

- The travel cost will be 11.

Combinatorial Problems

- Finding a combinatorial object - such as a permutation, a combination or a subset - that satisfies certain constraints and has some desired property.
- Examples
  - TSP, knapsack problem, job assignment problem, etc.
- Most difficult problems in computing.
  - The number of combinatorial objects typically grows extremely fast.

Geometric Problems

- Deal with geometric objects such as points, lines, and polygons.
  - Many applications in computer graphics, robotics, and tomography.
- Closest-pair problem: Given n points in the plane, find the closest pair among them.
- Convex-hull problem: Find the smallest convex polygon that would include all points of a given set.

Numerical Problems

- Originally, a computer was developed to solve many mathematical problems such as solving equations, computing definite integrals, etc.
  - However, the focus has shifted to business applications.
  - They are still important.

<<< Course Instruction >>>

- **Read pages 18 through 23** of the textbook.
  - It briefly introduces the important problem types in algorithms.
  - Some problems will be covered in subsequent lectures to illustrate different algorithm design techniques and methods of algorithm analysis.
  - You don’t need to understand all problem types in detail at the moment. However, you should read the section to understand the basic ideas of them.

<<< Course Instruction >>>

- This lesson is over.
  - If you have any questions, please contact your instructor.
- When you are done, study the next lecture (week_1_3.ppt) on the Canvas.