00-Introduction

August 7, 2020

1 CS3: Intro to Algorithms & Applications using C++

1.1 Table of Contents

1.2 Introduction

https://opendsa-server.cs.vt.edu/ODSA/Books/CS3/html/IntroDSA.html

- Some problems to think about…
  1. What is the fastest route from Los Angeles, CA to New York City, NY?
     - how long does it take to get there?
  2. What is the most reliable source on the Internet to learn about data structures and algorithms?
  3. Who was the most influential celebrity/politician of last year?
  4. What is an average salary of a software engineer?
  5. What is the cheapest way to travel from Grand Junction, CO, to Kathmandu, Nepal?
  6. How does Apple's Siri know what appointment you have next when you ask it?

1.3 Computer Science (CS) fundamentals

- the core of CS is representing data so that it can be efficiently stored and retrieved
  - many computer programs sole functionality is to do just that, but as fast as possible. e.g., search engines like, Google, Bing, etc.
- some programs may do heavy mathematical computation as fast as possible, e.g., wolframalpha computational intelligence (https://www.wolframalpha.com)
  - find factorial(10000) or 10000!

the study of data structures and the alogrithms that manipulate them to solve a given problem in feasible time and resource is the heart of compuer science

1.4 Goals of Data Structure (DS) and Algorithm courses

1. present a collection of commonly used data structures and algorithms, programmer's basic "toolkit"
   - for many problems these toolkit out-of-the box will provide efficient solutions
   - "toolkit" forms the basis/foundations to build more advanced data structures
2. introduce the idea of trade-offs between the costs and benefits associated with every data structure or algorithm, e.g., trade-offs between memory and time
3. learn to measure the effectiveness of a data structure or algorithm so you can pick the right ones for the job
   - also allows you to quickly judge the merits of new data structures or algorithms that you or others might invent

1.5 Solving problems

- the primary job of computer scientists is to solve problems!
- there are often too many approaches to solve a problem
- at the heart of computer program designs are two (sometimes conflicting) goals:
  1. to design an algorithm that is easy to understand, code, and debug
  2. to design an algorithm that makes efficient use of the computer's resources
     - "elegant" solutions meet both these goals!
     - software engineering focuses on goal 1, though we emphasize it from CS1!
     - CS2 and CS3 usually focuses on goal 2.

1.6 Why faster DS and algorithms when we've faster computers?

- according to Moore's law https://en.wikipedia.org/wiki/Moore%27s_law, no. of transistors in computer's circuit board doubles every two years.
- so, if processor speed and memory size continue to improve, won't today's hard problem be solved by tomorrow's hardware?
- additional computing power enables us to tackle more complex problems, such as sophisticated user interfaces such as in mobile devices, bigger problem sizes (big data), or new problems previously deemed computationally infeasible
- resources are always limited…
- efficient solution solves the problem within the required *resource constraints*.
  - may require fewer resources than known alternatives, regardless of whether it meets any particular requirements
- **cost** of a solution is the amount of resources that the solution consumes

  - measured typically in terms of one key resource such as time implying it meets all other resource constraints
  - e.g., fastest solutions on open.kattis.com problems meeting memory requirements: https://open.kattis.com/problems/cd/statistics

1.7 Selecting a Data Structure

1. analyze your problem to determine the **basic operations** e.g., inserting data item into the data structure, deleting, finding a specified data item

- quantify the resource constraints for each operation
- select the data structure that best meets these requirements

1.7.1 Some questions to think about to determine the importance of operations

1. is the application static or dynamic
   - in static applications, data is loaded at the beginning and new data are not inserted
   - in dynamic applications, new data items are inserted and may be inserted in any locations

- can data items be deleted? this may make the implementation more complicated
- how are the data items processed? in some well-defined order, random access?

1.8 Exercises

1. Which of these is NOT a definition for efficiency in a computer program?
   - it solves the problem within the required resource constraints
   - it requires fewer resources than known alternatives
   - it runs in linear time

[ ]: