CS240 Notes

Jacky Zhao

June 27, 2020

1 Course Objectives

1.1 Overview

What is this course about?

- When first learning to program, we emphasize correctness
- Starting with this course, we will also be converned with efficiency
- We will study efficient methods of storing, accessing, and performing operations on large collections of data.
- Typical operations include: inserting new data items, deleting data items, searching for specific data items, sorting
- We will consider various abstract data types (ADTs) and how to implemnet them efficiently using appropriate data structures.
- There is a strong emphasis on mathematical analysis in the course
- Algorithms are presented using pseudocode and analyzed using order notation (big-O, etc.)

Course Topics:

- big-O analysis
- priority queues and heaps
- sorting, selection
- binary search trees, AVL trees, B-trees
- skip lists
- hashing
- quadtrees, kd-trees
- range search
- tries
- string matching
- data compression

Required knowledge:

- arrays, linked lists (3.2- 3.4)
- strings (3.6)
- stacks, queues (4.2 - 4.6)
- abstract data types (4 - intro, 4.1, 4.8 - 4.9)
- recursie algorithms (5.1)
- binary trees (5.4 - 5.7)
- sorting (6.1 - 6.4)
- binary search (12.4)
- binary search trees (12.5)
- probability and expectations

1.2 General Terminologies

The core of CS240 is:

> Given problem Π, design algorithm A that solves it, and analyze its efficiency

So what is a problem, an algorithms, and how do you quantify efficiency?

Problem

- Given a problem instance, carry out a particular computational task
- Ex. Sorting is a problem

Problem Instance

- Input for the specified problem

Problem Solution

- Output (correct answer) for the specified problem instance

Size of a problem instance

- $Size(I)$ is a positive integer which is a measure of the size of the instance I

Algorithm

- a step-by-step process (e.g. described in pseudocode) for carrying out a series of computations, given an arbitrary problem instance I

Algorithm solving a problem

- an algorithm A solves a problem Π if, for every instance I of Π, A finds (computes) a valid solution for the instance I in finite time

Program

- an implementation of an algorithm using a specified computer language

Pseudocode

- a method of communicating an algorithm to another person
- in contrast, a program is a method of communicating an algorithm to a computer
- General rules of pseudocode:
  - omits obvious details (variable declarations)
  - has limited, if any, error detection
  - sometimes uses English descriptions
  - sometimes usus mathematical notation

1.3 Algorithms and programs

For a problem Π, we can have several algorithms.
For an algorithm A solving Π, we can have several programs (implementations)

Algorithms in practice: Given a problem Π:

1. **Algorithm Design:** Design an algorithm A that solves Π
2. **Algorithm Analysis:** Assess correctness and efficiency of A
3. If acceptable (correct and efficient), implement A.