Coding Interview: Algorithms and Data Structures

In this course you will learn data structures and algorithms by solving practice problems. You will begin each course by learning to solve defined problems related to a particular data structure and algorithm. By the end of each course, you would be able to evaluate and assess different data structures and algorithms for any open-ended problem and implement a solution based on your design choices.

An algorithms is, essentially, a well-defined set of rules or instructions that allow solving a computational problem. The theoretical study of the performance of the algorithms and the resources used by them, usually time and space, allows us to evaluate if an algorithm is suitable for solving a specific problem, comparing it with other algorithms for the same problem or even delimiting the boundary between viable and impossible.

Main References

This is a restricted list of various interesting and useful books that will be touched during the course. You need to consult them occasionally.

- Gayle Laakmann McDowell **Cracking the Coding Interview**
- Alexander S. Kulikov and Pavel A. Pevzner **Learning Algorithms Through Programming and Puzzle Solving**
- Aditya Y. Bhargava **Grokking Algorithms: An illustrated guide for programmers and other curious people**
- Thomas H. Cormen, Charles E. Leiserson **Introduction to Algorithms**
- Adnan Aziz, Tsung-Hsien Lee, and Amit Prakash, **Elements of Programming Interviews in Java: The Insiders' Guide**

Complementary References

- Ace the Coding Interviews
- Data structures in python an interview refresher
- Online Programming Learning Platform

Objectives

This course is primarily designed to boost your knowledge in algorithms and data structures by closing technical preparation gaps. This course if mainly for current software engineers with some years of experience.

Learning Ourcomes

- Develop the ability to evaluate the complexity and quality of algorithms proposed for a given problem
- Study the most representative, introductory algorithms of the most important classes problems threated in computation
- Develop the ability to solve algorithmic problems using the fundamental principles of algorithms design
- Be able to answer the following questions when a new algorihtm is presented. How good is the performance?, Is there a better way to solve the problem?

Prerequisites

An understanding of logic, maths, algorithms, and proficiency in some programming language is assumed.

Course Outline

Unit 01: Introduction to Complexity

Goals

- Differences among best, expected, and worst case behaviours of an algorithm
- Asymptotic analysis of upper and expected complexity bounds
- Big O notation: formal definition
- Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential
- Big O notation
- Analysis by recursion tree

#	Type	Format	Duration	Topic
00	reading	pre-work	30min	Introdution to complexity
01	reading	pre-work	30min	Asymptotic analysis
02	seminar	guided	10min	Selection sort
03	seminar	guided	10min	Merge sort
04	homework	self-placed	20min	Counting inversions

Unidad 02: Complexity in Random Algorithms

Goals

- Recurrence relations
- Master Theorem
- Analysis of iterative and recursive algorithms

#	Type	Format	Duration	Topic
00	reading	pre-work	30min	Complexity analyis
01	homework	self-placed	15min	Binary Search
02	reading	pre-work	30min	Random algorithms
03	seminar	guided	10min	Quick sort
04	seminar	guided	10min	Quick select
05	homework	self-placed	20min	Median of mediam

Unidad 03: List data structures

Topics

- Strings
- Arrays
- Linked Lists
- Queues and Stacks

#	Type	Format	Duration	Topic
00	reading	pre-work	5min	Strings
01	reading	pre-work	5min	Arrays
02	seminar	guided	10min	Valid Anagram problem
03	seminar	guided	10min	Intersection of two arrays
04	homework	self-placed	20min	Largest Range
05	homework	self-placed	20min	Group Anagrams
06	reading	pre-work	5min	Lists
07	seminar	guided	10min	Remove Kth node from end
08	homework	self-placed	10min	Merge two linked lists
09	reading	pre-work	20min	Stacks and queues
10	seminar	guided	20min	Balanced brackets

More problems

- Rotate Array
- Subarray sort
- Underscorify substrings
- Find loop

Unidad 04: Multi-way data structures

Topics

- Binary trees and Binary search trees (BSTs)
- Sets and Diccionaries
- Priority queues
- Hash tables

#	Type	Format	Duration	Topic
00	reading	pre-work	10min	Binary trees and BTSs
01	seminar	guided	10min	Find closest value in a BST
03	homework	self-placed	10min	Validate BST

#	Type	Format	Duration	Topic
04	reading	pre-work	10min	Sets and Diccionaries
05	reading	pre-work	10min	Priority queues
06	reading	pre-work	10min	Hash tables
07	seminar	guided	15min	Two sum problem

More problems

- Branch Sums
- Max Path Sum in Binary Tree
- Find Closest Value in BST
- Validate BST
- Same BSTs
- Continuous median

Unit 05: Algorithms Strategy

Topics

- Brute Force
- Recursion and Divide-and-conquer
- Dynamic Programming

#	Type	Format	Duration	Topic
00	reading	pre-work	10min	Recursion and divide and conquer
01	reading	pre-work	10min	Dynamic Programming
02	seminar	guided	10min	Fibonacci
02	seminar	guided	15min	Hannoi
03	seminar	guided	15min	Number of Ways to Make Change

More problems

- Min number of coins for change
- Min number of jumps
- Knapsack problem
- Longest String Chain

Unit 06: Graphs Algorithms

Topics

- Graph representation
- Searching algorithms

- Shortest Path

#	Type	Format	Duration	Topic
00	reading	pre-work	30min	Graphs and representation
01	reading	pre-work	30min	Searching Algorithms
02	seminar	guided	10min	Depth-first Search
03	seminar	guided	10min	Breadth-first Search
04	homework	self-placed	20min	Simple shortest Path
05	seminar	guided	15min	Bellman Ford algorithm

More problems

- Airport Connections
- Number of islands
- Youngest Common Ancestor
- River Sizes

Author

- Alexander Ocsa