

Who should take this course?

Thanks for choosing this course! We hope the techniques you learn here go a long way towards building solid algorithmic skills.

Intended Audience

If you want to work on your algorithms and problem-solving skills, this course is for you. In a systematic way, we will start with easier problems to build an understanding of the underlying solution patterns and then apply these patterns to solve difficult ones. Every problem presented in this course has been solved in **Java**, **Python**, **C++**, and **JavaScript**.

Prerequisites

You need to have a basic understanding of common data-structures like Array, LinkedList, HashMap, Stack, Queue, Heap, and Graph. You should also be familiar with [Recursion](#) and [Big-O](#).

Course Overview

This course categorizes coding interview problems into a set of **16 patterns**. Each pattern will be a complete tool - consisting of data structures, algorithms, and analysis techniques - to solve a specific category of problems. The goal is to build an understanding of the underlying pattern so that we can apply that pattern to solve other problems.

We have chosen each problem carefully such that it not only maps to the same pattern but also presents different constraints. Overall, the course has around **150 problems** mapped to 16 patterns.

The problems solved under these patterns use a varied set of algorithmic techniques. We will make use of **Breadth-First Search** and **Depth-First Search** to solve problems related to **Trees** and **Graphs**. Similarly, we will also cover **Dynamic Programming**, **Backtracking**, **Recursion**, **Greedy algorithms**, and **Divide & Conquer**.

We will start with a brief introduction of each pattern before jumping onto the problems. The first problem, under each pattern, will explain the underlying pattern in detail to build the concepts that can be applied to later problems. The later problems will focus on the different constraints each problem presents and how our algorithm needs to change to handle them.

Let's start with **Sliding Window** pattern.