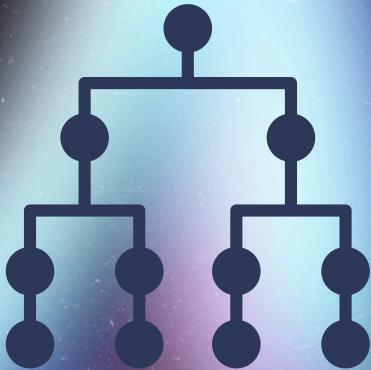


The ultimate

Data structures and Algorithms Study guide



Lucas Germinari

Introduction

Programming is all about data, we all know that by now. The instructions of our programs are stored in the same way as any data in the computer memory, items in collections are stored in one or more data structures, a stack is used to implement a call stack, and... well, you got the idea. Those examples illustrate part of the relevance of data structures – different logical ways to store and organize data – and algorithms – the sequence of steps that define the operations made on data.

The importance of properly applying DSA cannot be overstated. In fact, for a very big application like Facebook, using the right data representations can make a simple operation of retrieving your list of friends (usually stored in a data structure called graph) take either less than a second or 16 minutes! It is hard to believe, right? Another use of DSA comes before you even start professionally working with them, during the job interview. This topic is easily among the most requested in technical assessments.

Those are the kinds of benefits that this study guide intends to help you achieve. It will provide you, using readily accessible links, with all the materials you need to begin with and master data structures and algorithms. It also indicates resources for practicing your skills – don't skip this part, believe me! Having that said, let's get started.

Getting ready

Before diving into DSA, I recommend that you acquire (or refresh) two other skills: programming and, optionally, some discrete math. Don't worry if you do not like math, you can skip that part and focus on programming.

Let's begin with the programming language. You can choose any programming language you would like or use the one(s) you already know. The concepts addressed by this guide are universal, and only some implementation details and syntax change for different languages. If you want some help to get started, here it is:

- Harvard's CS50X: a free course on computer science and programming that mainly uses C, but also introduces Python.
 - <https://cs50.harvard.edu/x/2024/>
- Harvard's CS50P: a full and free Python programming course.
 - <https://cs50.harvard.edu/python/2022/>
- If you are looking for other languages, FreeCodeCamp has a ton of resources.

Now, let's take care of the math, which is an **optional part**. MIT is known for its excellence, and its course on Math for Computer Science course could not be different. It teaches many things beyond what is needed for DSA, but those topics would be incredibly useful for computing in general.

- <https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/>

If you want to go faster and pick up only the essentials, I would go with the amazing FreeCodeCamp course about Big O analysis:

- <https://www.youtube.com/watch?v=Mo4vesaut8g>

Dive deep into DSA

Now, it's time to actually learn data structures and algorithms! I mentioned MIT in the last section, and here are they again! Their freely available course "Introduction to Algorithms" will teach you everything you need to know about DSA in depth:

- <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/resources/lecture-21-algorithms2014next-steps/>

I should say, however, that this course is more theoretical than other materials linked here and involve more math, but that has its benefits. It will empower you properly analyze algorithms and run times, what in turn is essential when choosing the correct structures to be used in each situation. But don't worry, they have tons of problem sets for practice.

Now, if you like more practical and implementation-focused approaches (while still covering the same content as MIT's course), I would recommend any of these materials on YouTube:

- In Python, from FreeCodeCamp
 - <https://www.youtube.com/watch?v=pkYVOMu3MgA>
- In C/C++, from FreeCodeCamp
 - <https://www.youtube.com/watch?v=B3lLgl4Y4DQ&pp=ygUtZnJIZSBjb2RlIGNhbXAgZGFOYSBzdHJ1Y3R1cmVzIGFuZCBhbGdvcmloaG1z>
- In Java, from Bro Code
 - <https://www.youtube.com/watch?v=CBYHwZcbD-s>

Practice, practice, practice...

The only way to get good at DSA is by practicing, so this section aggregates the best resources for effectively solving programming problems that involve algorithms and data structures. To begin with, the most famous platform for this purpose, LeetCode (and it's free):

- <https://leetcode.com/>

Besides having an enormous variety of well-developed coding problems, LeetCode has resources and playlists that focus on technical interview preparation so you can that job position!

This free website has a series of problems for practice, which go from easiest to hardest to provide you with a seamless learning experience:

- <https://www.hackerrank.com/domains/algorithms>

Now, if you are willing to pay a subscription for a learning platform, I would recommend Brilliant. They have tons of visually appealing interactive lessons and practices.

- <https://brilliant.org/courses/computer-science-algorithms/>

Another great way to learn data structures is by implementing them yourself. For this purpose, I coded many data structures in C++ and made them available in this GitHub repository:

- <https://github.com/Germinari1/DS-PlusPlus>

Feel free to take a look at the implementations, play around, and try to do it yourself.

Your next steps

By now you **learned a lot**, wow! As always, however, you can go deeper into the current topic and explore more advanced stuff. My first recommendation is this free FreeCodeCamp course on dynamic programming:

- <https://www.youtube.com/watch?v=oBt53YbR9Kk&pp=ygUiZnJlZSBjb2RIIGNhbXAgZHluYW1pYyBwcm9ncmFtbWluZw%3D%3D>

You might also be interested in MIT's free course on design and analysis of algorithms, which will empower you to create your own efficient algorithms!

- https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/video_galleries/lecture-videos/

Finally, I recommend this Harvard's collection of lectures on advanced algorithms and analysis:

- <https://www.youtube.com/watch?v=OJUN9aDxVmI&list=PL2SOU6wwxB0uP4rJgf5ayhHWgw7akUWSf>

Use your knowledge

After learning all that stuff (wow, it was a lot of knowledge), you are prepared to apply data structures and algorithm concepts to real-world problems and applications. Here are some ideas:

- Crack technical interviews
- In your next application, use a custom-implemented data structure
- When choosing an algorithm or data structure for your next project, take into consideration asymptotic analysis, space-time complexity trade-offs, and the benefits and drawbacks of each option in the context of your specific application requirements.