UDACITY

NANODEGREE PROGRAM SYLLABUS

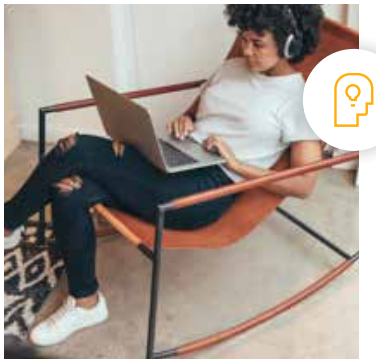
Data Structures & Algorithms

Overview

In this course you will learn data structures and algorithms by solving 80+ 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.

Estimated Time: 4 Months at 10hrs/week

Prerequisites: Python and Basic Algebra



Flexible Learning: Self-paced, so you can learn on the schedule that works best for you

Need Help?
udacity.com/advisor
Discuss this program with an enrollment advisor.

Course 1: Supervised Learning

Get an overview of your program. Meet your instructors, and refresh your python skills. Learn the framework to deconstruct any open-ended problem and then understand the concepts of time and space complexity, essential tools for evaluating different data structure & algorithms.

Course Project : Unscramble Computer Science Problems

Deconstruct a series of open-ended problems into smaller components (e.g, inputs, outputs, series of functions).

	LEARNING OUTCOMES
LESSON ONE	Introduction
LESSON TWO	Python Refresher
LESSON THREE	How to Solve Problems
LESSON FOUR	Big O Notation

Course 2: Data Structures

Learn different data structures that can be used to store data. Implement different methods used to manipulate these data structures and examine the efficiency. Understand the advantages and applications of different data structures. Learn how to approach open ended problems (either in interviews or in real-world scenarios) and select appropriate data structures based on requirements.

Course Project
Show Me the Data Structures

Solve a series of open-ended practice problems such as LRU Cache, Private Blockchain, File Recursion and many more. Hone your skills to identify and implement appropriate data structures and corresponding methods which meet given constraints.

	LEARNING OUTCOMES
LESSON ONE	Collection data structures (lists, arrays, linked lists, queues, stack)
LESSON TWO	Recursion
LESSON THREE	Trees
LESSON FOUR	Maps and Hashing

Course 3: Basic Algorithms

Learn and implement basic algorithms such as searching and sorting on different data structures and examine the efficiency of these algorithms. Use recursion to implement these algorithms and then learn how some of these algorithms can be implemented without recursion. Practice selecting and modifying these algorithms for a variety of interview problems.

Course Project
Problems vs. Algorithms

A series of real-world open ended problems such as request routing for web server, search-term auto-completion and Fibonacci heap which train you to apply suitable data structures and algorithms under different context.

	LEARNING OUTCOMES
LESSON ONE	Binary Search
LESSON TWO	Sorting Algorithms
LESSON THREE	Divide & Conquer Algorithms

UDACITY

Course 4: Advanced Algorithms

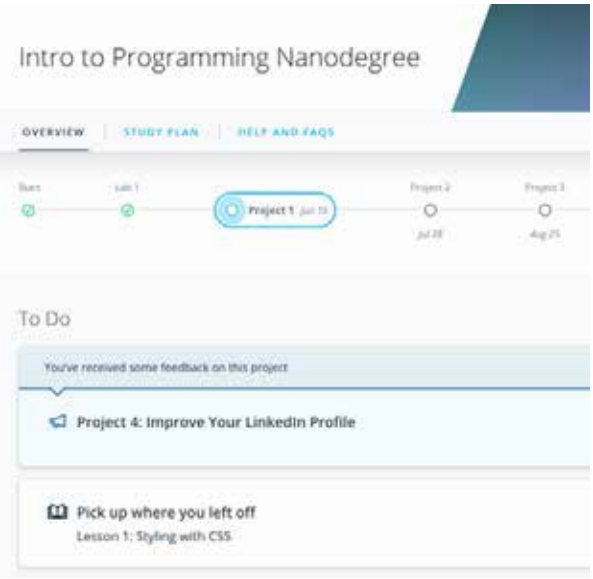
Build on your algorithm skills by learning more advanced algorithms such as brute-force greedy algorithms, graph algorithms, and dynamic programming, which optimizes recursion by storing results to sub problems.

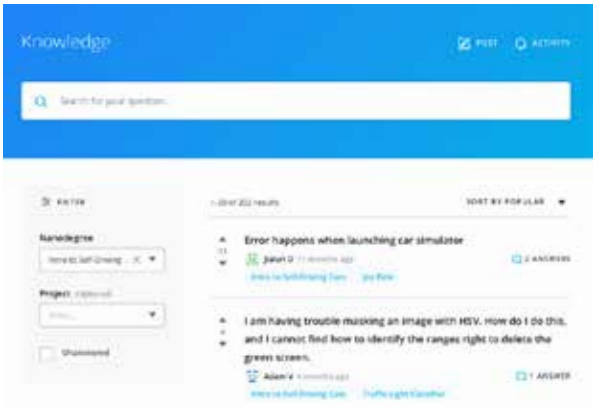
Course Project
Route Planner

In this project, you will build a route-planning algorithm like the one used in Google Maps to calculate the shortest path between two points on a map. You will first select and implement appropriate data-structure to represent points on a map and then implement the A* algorithm to find shortest path.

	LEARNING OUTCOMES
LESSON ONE	Greedy Algorithms
LESSON TWO	Graph Algorithms
LESSON THREE	Dynamic Programming
LESSON FOUR	Linear Programming

Our Classroom Experience





REAL-WORLD PROJECTS
Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE
Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students and discover in real-time how to solve the challenges that you encounter.

STUDENT HUB
Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with your technical mentor and fellow students in your Nanodegree program.

WORKSPACES
See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES
Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS
Work with a mentor to create a custom study plan to suit your personal needs. Use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER
Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Brynn Claypoole

INSTRUCTOR

Brynn is a former Udacity employee who worked as Lead Data Analyst at Udacity before joining Facebook as Data Engineer. Currently, she is working as software engineer with 10x Genomics.

Abe Feinberg

CONTENT DEVELOPER

Abe is a Content Developer at Udacity and previously taught university courses in psychology and computer science. He loves both learning and teaching, and has a particular passion for breaking down difficult concepts and making them easier to master.

Kyle Stewart-Franz

CONTENT DEVELOPER

Kyle has developed projects for a variety of Udacity's Nanodegree programs, such as Self-Driving Car Engineer, Robotics, and Blockchain. Kyle, a self-taught developer, is always striving towards creating great learning experience for students.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW

WHY SHOULD I ENROLL?

Whether you want to be a web developer, a machine learning engineer, or a data scientist, having a deep understanding of Data Structures and Algorithms is essential to acing job interviews and becoming a successful software engineer. As Linus Torvalds famously said, "Bad programmers worry about the code. Good programmers worry about data structures and their relationships".

The Data Structures and Algorithms Nanodegree program will help you excel at solving everything from well-defined problems, like how to calculate the efficiency of a specific algorithm, to more open-ended problems, like building your own private blockchain or writing a web-crawler.

You'll work on over 80 exercises and four real-world projects so that you can get the hands-on practice required to learn how to implement appropriate solutions based on your design choices.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

While this course is not designed to prepare you for a specific job, after completing this program, you will have had extensive practice solving data structures and algorithm problems to help you prepare for the data structures and algorithms part of coding interviews.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you are looking to improve your skills in data structures and algorithms -- to prepare for the technical portion of job interviews, to improve your software engineering skills, etc. -- then this Nanodegree program will provide you with extensive practice with defined and open-ended problems so that you learn how to implement the appropriate solution based on your design choices.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

To optimize your chances of success in the Data Structures and Algorithms Nanodegree program, you should have the following knowledge:

- Intermediate Python programming
- Basic algebra

FAQs Continued

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

If you are new to programming, we recommend the **Introduction to Programming Nanodegree** program.

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Data Structures and Algorithms Nanodegree program is comprised of content and curriculum to support four (4) projects. We estimate that students can complete the program in four (4) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

- Python 3
- A code/text editor, such as vim, Sublime Text, Atom, or VSCode
- A web browser
- A command line interface, such as Terminal (on Mac) or Git Bash (on Windows)

Hardware Requirements:

A modern personal computer running macOS, Windows, or Linux, with a high-speed Internet connection.