



Computer Science

1. Intro to Python and OOP
2. Algorithms
3. Data Structures
4. CS Unit 1 Build
5. Hash Tables and Blockchain
 - Arrays
 - Hash Tables
 - Blockchain I - Structure
 - Blockchain II - Distribution
6. Graphs
7. Computer Architecture
8. CS Unit 2 Build

Hash Tables

Hash tables are arguably the single most important data structure known to developers. Used to implement everything from objects in JavaScript and dictionaries in Python to Memcached over a distributed computer network, hash tables are beloved by programmers for providing key/value storage with constant big-O time complexity for insertion, deletion and search.

We will be looking underneath the hood at how this delightful data structure works and answering questions like, how do hash tables work? What is a hash function? What is a hash collision and how are they handled? What is the “birthday problem”? What do we do when our hash table is full?

You can think of hash tables as extensions of arrays. In class, we will implement arrays and compare them to linked lists. For your project, you will use this knowledge to implement hash tables in Python.

At first glance, this may seem like a silly task. After all, both of these data structures are already built into Python and their implementations are fast, full of features, and of course, already working.

The reason we are doing this isn't so that when we're done we will *have* an implementation of these data structures. We do it so that we can obtain a deeper *understanding* of how our predecessors have invented elegant solutions to difficult problems so that we are better equipped to solve the problems we will encounter on the job.

At the end of this module, you should be able to:

- implement and utilize a basic hash table.
- handle collisions and resizing in a hash table.

Pro Tip

Prepare

Review each preclass resource before class.

Hash Tables - Intro



Hash Tables - Collisions and Resizing



[Hash Tables](#)

Github Repo with Hash Table overview

Learn

Learn to implement and utilize a basic hash table.

How do hash tables work? By writing a basic implementation, you will be able to see the strengths and limitations of a hash table. How much space does an empty hash table occupy in memory? What role does the hash function play? How does a hash table achieve the rare and mythic property of constant time complexity for insertion, deletion and search?

Overview

What are hash tables? What is a hashing function? What is the performance of a hash table?

Follow Along

We will go over basic functions of a hash table and hash functions using diagrams and pseudo-code.

Challenge

Implement a hash table (no collision handling) in Python, without using any built-in functions to help.

Dig Deeper

- [Hash Tables](#)
Github Repo with Hash Table overview

Learn to handle collisions and resizing in a hash table.

In your initial implementation of a hash table, you may have come across the problem: What happens when two keys hash to the same index? We will be discussing two methods of resolving hash collisions: open addressing and linked list chaining.

With both of these methods, the hash table's performance will degrade as it fills up. We will discuss resizing the hash table, including when to resize and how this operation affects the hash table's big-O.

Overview

We will be discussing various methods to handle hash table collisions

Follow Along

We will be discussing various methods to handle hash table collisions.

Challenge

Implement linked list chaining in your basic hash table.

Project

[Hash Tables](#)

Hash tables are arguably the single most important data structure known to mankind. Used to implement everything from objects in JavaScript and dictionaries in Python to Memcached over a distributed computer network, hash tables are beloved by programmers for providing key/value storage with constant big-O time complexity for insertion, deletion and search.

Review

Class Recordings

You can use class recordings to help you master the material.

[Hash Tables for CSPT4 with Tim Roy](#)

Hash Tables for CSPT4 with Tim Roy

[All previous recordings](#)

Demonstrate Mastery

To demonstrate mastery of this module, you need to complete and pass a code review on each of the following:

- Objective challenge:
Implement a hash table (no collision handling) in Python, without using and built in functions to help.
- Objective challenge:
Implement linked list chaining in your basic hash table.
- Project: Hash Tables