



What is a Hash Table?

This lesson is a brief introduction to the hash table data structure and the basic principle behind implementing it.

We'll cover the following



- Hashing
- Hash Tables

Hashing

Until now, the overall time complexity accomplished by most of the data structures in **insertion**, **deletion**, and **search** was up to $O(\log n)$ or $O(n \log n)$, which is pretty good. But for a significantly large amount of data, this complexity starts to adversely affect the efficiency of an algorithm.

The ideal data structure is one that takes a constant amount of time to perform all three operations. And that is where **hashing** steps into the spotlight!

Hashing is a process used to store an object according to a unique key. This means that hashing always creates a **key-value pair**. A collection of such pairs forms a *dictionary* where every object or value can be looked up according to its key. Hence, the search operation can be performed in $O(1)$.

The concept of hashing has given birth to several new data structures, but the most prominent one is the **hash table**.



Hash Tables



If your algorithm prioritizes search operations, then a hash table is the best data structure for you. In Python, hash tables are generally implemented using lists as they provide access to elements in constant time.

In Python, we have several in-built types such as `set` and `dict` which can provide us the hash table functionality.

3	<key>	<data>
⋮		
16	<key>	<data>
17	<key>	<data>

In this section, we will implement our own hash table since it is a very popular topic in coding interviews.

Let's get started by defining the building blocks on an efficient hash table.

The performance of a hash table depends on three fundamental factors:

- Hash function
- Size of the hash table
- Collision handling method

We will study all three concepts in the following lessons.

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. [See how](#) ⓘ



[← Back](#)

The Hash Function

☒ Mark as Completed Report an Issue