

Day 29 – Hashing

Chapter 08, 8.4 – Hashing, Review Problem(s): **15**

Chapter 08, 8.5 – Hash Functions, Review Problem(s): **20**

Chapter 08, 8.6 – A Hash-Based Map, Review Problem(s): **25**

Chapter 08, 8.7 – Map Variations

Today we will explore an efficient approach to implementing a Map, called Hashing.

Recall that the key-value pairs can be considered mathematical functions:

- Only one key is allowed.
- But different keys can point to the same value.

For example, $f(x) = x^2$, both $f(-2)$ and $f(2)$ are equal to 4.

In hashing, if two key values point (“compress”) to the same location, we call it a **collision**. Minimizing collisions is the biggest challenge in designing a good hashing system.

Chapter 08 PPT, pg 16-23.

Hash Lab handout.

There are many factors affecting the performance of Hashing and minimizing collisions.

Chapter 08 PPT, pg 24-29.

Some programming languages directly support Hashing, and many other languages, provide map functionality through their standard code libraries.

Maps are known by many names:

- **Symbol table** - one of the first carefully studied and designed data structures, and were related to compiler design
- **Dictionary** - the idea of looking up a word (the key) in a dictionary to find its definition (the value) makes the concept of a dictionary a good fit for maps
- **Hashes** - because a hash system is a very efficient and common way to implement a map, you will sometimes see the two terms used interchangeably

- **Associative Arrays** - You can view a map as an array—one that associates keys with values rather than indices with values.

Chapter 08 PPT, pg 31-32.