Cheatsheet - Hash Tables

Fabio Lama – fabio.lama@pm.me

1. About

A hash table is a data structure that maps keys to values using a hash function. This function transforms the input (key) into a fixed-size integer, which serves as an index in an array, enabling fast data retrieval. Hash tables are widely used due to their average $\Theta(1)$ time complexity for both insertion and lookup operations.

2. Search Algorithms Complexity

> **WARNING** This table assumes no collisions in the hash table.

Name	Worst case (time)	Best case (time)	Worst case (space)	Best case (space)
Linear Search	$\Theta(N)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
Binary Search (iterative)	$\Theta(\log N)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
Binary Search (recursive)	$\Theta(\log N)$	$\Theta(1)$	$\Theta(\log N)$	$\Theta(1)$
Direct Addressing	$\Theta(1)$	$\Theta(1)$	$\Theta(k)$	$\Theta(k)$
Hash Table	$\Theta(1)$	$\Theta(1)$	$\Theta(M)$	$\Theta(M)$

Where k is the maximum possible key value and M is the size of the hash table.

3. Hash Tables

Hash tables use an index of an array to represent a number. Consider an array of size 7 and the follwing, simple hash function:

$$h(x) = x \bmod 7$$

For example, lets say $x = 11$ and we compute:

$$h(11) = 11 \bmod 7 = 4$$

This means the number 11 is stored in the array at index 4. Knowing the index, the search for that number is very fast. However, given that the size of this array is very small, **the number of collisions can be very high**, depending on number of inputs.

For example, both 11 and 18 would be stored at index 4:

$$h(11) = 11 \bmod 7 = 4$$

$$h(18) = 18 \bmod 7 = 4$$

If necessary, the hash table can be **extended** with a larger array, but this requires rehashing all the existing elements. Alternatively, the method of **linear probing** can be applied to use the next available index in case of a collision (note that this information must be stored somewhere). Or, each index can be a pointer to a separate, nested table, which is a **separate chaining** method.

The best case and worst case complexity for hash tables must consider those collision handling methods as well, as in how it behaves with no collisions at all and with all elements colliding, respectively.

Last updated 2024-06-17 15:21:54 UTC