

Discussion Week 6 - Hash Tables

Last updated Spring 2018

Topics Covered

- Dictionaries and assoc lists
- Hash tables
- Chaining
- Open addressing
 - Linear probing
 - Quadratic probing
 - Double Hashing
- Methods of hashing
- Hash table performance

Mini Lecture

A hash table is a kind of a dictionary (that is, a mapping from arbitrary keys to arbitrary values) that uses a special function to calculate where in its available cells it will store each key/value pair. This special function is called a *hash function*.

Hash tables are particularly useful because they have approximate $O(1)$ performance on storage and lookup, which seems like black magic. Of course, they make up for this by being space inefficient most of the time, but hey - no free lunch.

Definitions

Dictionary: a data structure that affords `Store(key, val)` and `Lookup(key)` methods, which do what they sound like. A dictionary is also called a *map*, *mapping*, or *associative array*.

Hash Table: a dictionary that stores its key/value pairs in an array, and which decides which array cell to use by running a *hash function* on the key.

Hash Function: a magic function for turning keys into valid array indices. *Note: by magic I do not mean difficult to understand - I mean unspecified and particular to each hash table.*

Collision: when two keys hash to the same array index, we refer to this as a collision.

Load Factor: called alpha, the load factor of a hash table is n/m , where n is the number of elements stored in the table, and m is the size of the array used for storage.

Chaining: one method of resolving collisions - simply store a small linked list in every array cell, and store elements at the end of the linked list. Simple, but not always very efficient.

Open Addressing: another method of resolving collisions. Instead of storing linked lists in the table,

we use a probing function to calculate the next available cell in the array (based off of the hash and some other variable), and store the element there.

Linear and Quadratic Probing: two methods of probing. Hard to unpack here. Ask your TA or check the slides.

Double Hashing: another method of probing. Uses a second, distinct hash function to complicate the result of the first hash. Helps to resolve collisions between elements that ordinarily hash to the same thing.

Worksheet

No worksheet for this week - it's midterm season.

Microquiz

1. Hash tables don't technically have $O(1)$ performance - they have $O(1 + a)$ performance, where $a = n/m$. Why do we still refer to this as constant time? (10 words max.)
2. Why are prime numbers so important for double hashing probe sequences? (20 words max.)
3. Describe the difference between a *perfect* hash and a *universal* hash. (20 words max.)