**Basics**

* Key value pairs.
* Whenever you create a key or use a key for lookup, an underlying hash function is responsible for decoding this key into a hash to find where in memory the computer needs to go. You want this decoder to be fast. You assume that this decoder has O(1) complexity.

**Hash Function**

* Give it an input, and it returns a hash number.
* Idempotent – A function given an input, will always output the same output. Results in extremely fast lookup.

**Hash Collisions**

* Insert, lookup, delete, and search are all O(1).
* Since assigning is random, and memory is limited, you will get situations where multiple key value pairs are stored in the same address space by the computer.
* Occasionally, lookup and writing can become O(n) because of collisions.
* There need to be ways to deal with these collisions.