Hash Table

A **hash table** (also called a **hash**, **hash map**, **map**, **unordered map** or **dictionary**) is a data structure that pairs keys to values.

```
C# (beta)
var lightBulbToHoursOfLight = new Dictionary<string, int>();

lightBulbToHoursOfLight.Add("incandescent", 1200);
lightBulbToHoursOfLight.Add("compact fluorescent", 10000);
lightBulbToHoursOfLight.Add("LED", 50000);
```

Hash tables:

- take *on average* $O(1)$ **time for insertions and lookups**
- are **unordered** (the keys are not guaranteed to stay in the same order)
- can use **many types of objects as keys** (commonly strings)

Hash tables can be thought of as arrays, if you think of array indices as keys!

In fact, hash tables are *built on* arrays. So if you ever want to use a hash table but know your keys will be sequential integers (like 1..100), you can probably save time and space by just using an array instead.

Note: hash tables have an **average case** insertion and lookup cost of $O(1)$. In industry, we often confuse the average-case cost with *worst case* cost, but they're not really the same. Because of hash collisions and rebalancing, a hash table insertion or lookup can cost as much as $O(n)$ time in the worst case. But usually in industry we assume hashing and resizing algorithms are clever enough that collisions are rare and cheap.