Hash Tables

Computing

How O(1) lookup actually works, and when it stops being O(1).

Every dictionary, map and set in every language is built this way.

1. **Hash the key**
   A function turns the key into a large integer, deterministically.

2. **Reduce to an index**
   Modulo the array size gives a slot.

3. **Handle collisions**
   Two keys can land in one slot — chain them, or probe for the next free one.

4. **Resize when full**
   Past a load factor around 0.7, allocate a bigger array and rehash everything.

WHEN IT DEGRADES

Many collisions → O(n) in the worst case
Resizing is O(n), but amortised across inserts
A mutable key changes its hash and becomes unfindable

> **Never mutate an object after using it as a key. Its hash changes and the entry becomes unreachable.**