

Hashmaps

Definition 0.1

Hashmaps: array backed data structure that allows us to use “custom” or flexible keys instead of using only indices. It uses a *hashing* function to dictate which indexes corresponds to what data in the backing array.

Theorem 0.1

Where to put Key?

`index = abs (key.hashCode() & arr.length)`

How do we avoid collisions?

We do this by controlling the size of the backing array. **For hashmaps, it is very bad to let the backing array get full**, because this causes collisions (from the % operator). In order to solve this, we resize while there is still space left in the array.

Theorem 0.2

$$\text{load factor} = \frac{\text{size}}{\text{capacity}}$$

We usually set a maximum allowed load factor in order to avoid collisions. This factor is typically between 0.6 to 0.8, and Java’s default is 0.75. After this loadfactor, we resize.

Note 0.1

Small Load Factors has more resizes and fewer collisions, while **Large Load Factors** are more efficient with memory but cause more collisions.

Another way to avoid collisions is to have a *good hashing function*, where different items have different hashes, and a harder goal: similar items have way different hashcodes.