

ALL ABOUT **HASH MAP**

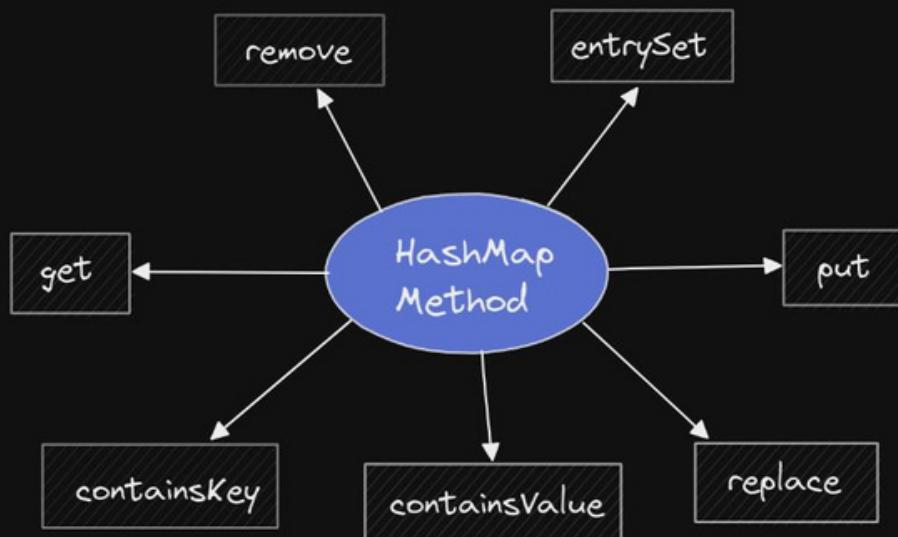


HashMap Notes:-

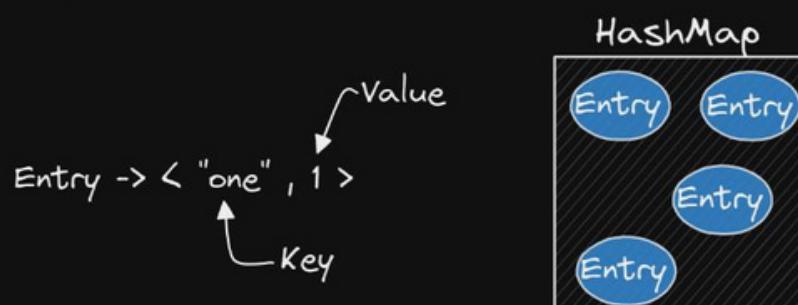
- Hashing is a technique which is used to efficiently find or store an item in a collection.
- Elements of Map are stored in key/value pairs.



- Remember that worst case complexity of hashing is still $O(n)$, but it gives $O(1)$ on the average.



- Entry is an object that maps keys to values in HashMap.



HashMap Internal-Working

- In Internal working of HashMap, we just need to divide that particular key to its length. through which we can get particular index in Hash-Table.

$$\text{Hash(Key)} = \text{Key \% 7}$$

That's How we Find Index in our Hash-Table, 7 is length of Keys.

$$\text{Keys} = [24, 16, 12, 17, 18, 9]$$

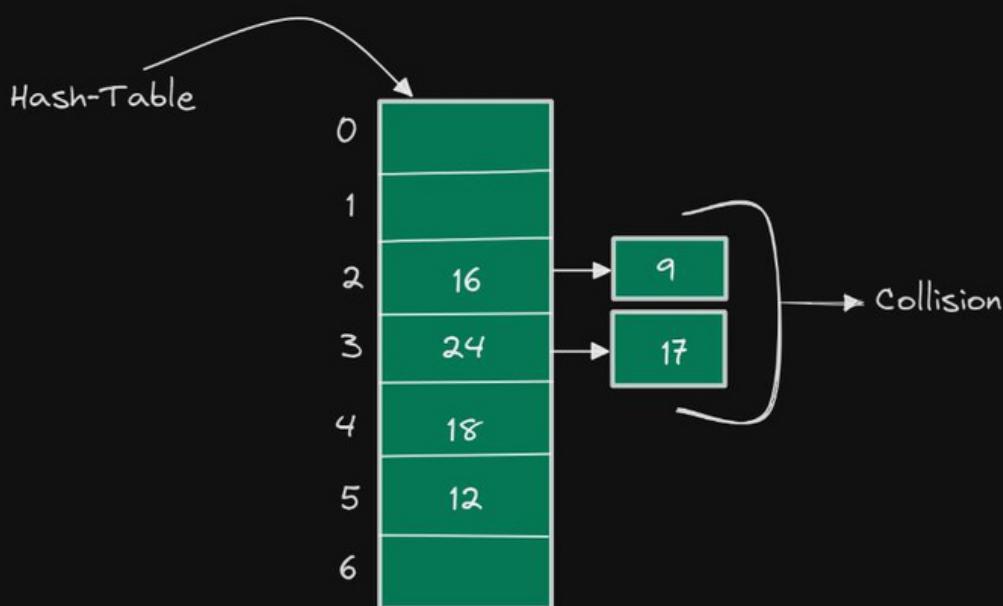
Given Keys we have to Place in Hash-Table.

$$\text{key \% 7}$$

$$\text{Hash-Table} \rightarrow [3, 2, 5, 3, 4, 2]$$

Here we get Index's of particular key In our Hash-Table.

- So, In Our Hash-Table put "KEY" at "Key \% 7".
Like -> 24 at 3.



HashMap Theory

◆ Hashing has four key components:

- 1) Hash Table
- 2) Hash Functions
- 3) Load Factor
- 4) Collisions

◆ **Hash Table**

- Hash table is a generalization of array.
whose key is k & position is kth in array is called direct addressing.
- Hash-table / hash-map is a data-structure that stores the keys and its values.
hash table uses a hash function to map keys to their associated values.

◆ **Hash Function**

- The hash function is used to transform the key into the index.
- Ideally, the hash function should map each possible key to a unique slot index.

◆ **Load Factor**

- This is the decision parameter used when we want to rehash or expand the existing hash table entries.

$$\text{Load Factor} = \frac{\text{Number of element in HashMap}}{\text{HashMap Size}}$$

◆ **Collisions**

- Collision is the condition where two records are stored in the same location.