

Assume you have studied java documentation on dictionaries methods like hashmap etc where you know all the names of methods then Provide all the methods you studied from beginner to advance without leaving any method where I will be satisfied by the response and can able to solve any leetcode problems

That's a great goal! Below is a **comprehensive list of `HashMap` methods** in Java, from beginner to advanced. This includes **every method** from the `HashMap` class (which implements `Map` interface), as well as useful methods inherited from `Map` and `AbstractMap`. With this list and practice, you'll be well-equipped for any Leetcode question involving `HashMap`.

---

## Beginner-Level Methods (Most Common)

Method	Description
<code>put(K key, V value)</code>	Adds or updates a key-value pair.
<code>get(Object key)</code>	Retrieves the value for the given key, or <code>null</code> if not found.
<code>containsKey(Object key)</code>	Returns <code>true</code> if the key exists.
<code>containsValue(Object value)</code>	Returns <code>true</code> if the value exists.
<code>isEmpty()</code>	Checks if the map is empty.
<code>size()</code>	Returns the number of key-value pairs.
<code>remove(Object key)</code>	Removes the entry for the given key.
<code>clear()</code>	Removes all entries from the map.



## Intermediate Methods (Used Frequently in Interviews)

Method	Description
<code>keySet()</code>	Returns a <code>Set</code> of all keys.
<code>values()</code>	Returns a <code>Collection</code> of all values.
<code>entrySet()</code>	Returns a <code>Set</code> of <code>Map.Entry&lt;K, V&gt;</code> for iteration.
<code>getOrDefault(Object key, V defaultValue)</code>	Returns the value or a default if key not found.
<code>putIfAbsent(K key, V value)</code>	Adds the entry only if the key is not already mapped.
<code>remove(Object key, Object value)</code>	Removes the entry only if key is mapped to the given value.
<code>replace(K key, V value)</code>	Replaces the value only if the key is mapped.
<code>replace(K key, V oldValue, V newValue)</code>	Replaces the value only if key is mapped to oldValue.

---



## Advanced Methods (Java 8+ Functional Style)

Method	Description
<code>forEach(BiConsumer&lt;? super K, ? super V&gt; action)</code>	Applies an action to each key-value pair.
<code>compute(K key, BiFunction&lt;? super K, ? super V, ? extends V&gt; remappingFunction)</code>	Recomputes the value for the key.
<code>computeIfAbsent(K key, Function&lt;? super K, ? extends V&gt; mappingFunction)</code>	Computes a value if key is not mapped.
<code>computeIfPresent(K key, BiFunction&lt;? super K, ? super V, ? extends V&gt; remappingFunction)</code>	Computes a value if key is already mapped.

```
merge(K key, V value, BiFunction<? super V,? super V,? extends V> remappingFunction)
```

Merges a value with the existing one.

```
replaceAll(BiFunction<? super K,? super V,? extends V> function)
```

Replaces each entry's value using the provided function.

---



## Inherited From **Map** Interface

These methods are implemented by **HashMap** as part of **Map**:

- `equals(Object o)`
  - `hashCode()`
  - `of(...)` (static factory methods in **Map** interface from Java 9)
  - `copyOf(Map<? extends K,? extends V> map)` – Java 10+
- 



## Utility Tips for Leetcode

**Frequency Count Template:**

```
Map<Integer, Integer> freq = new HashMap<>();  
freq.put(num, freq.getOrDefault(num, 0) + 1);
```

1.

**Iterating Through Map:**

```
for (Map.Entry<Integer, Integer> entry : map.entrySet()) {  
    int key = entry.getKey();  
    int value = entry.getValue();  
}
```

2.

### Sorting Map by Value or Key (via Streams):

```
map.entrySet()
    .stream()
    .sorted(Map.Entry.comparingByValue()) // or comparingByKey()
    .forEach(entry -> System.out.println(entry.getKey() + ": " + entry.getValue()));
```

3.

---

### Internal Behavior (For Advanced Understanding)

- Backed by a hash table (buckets + linked list or tree).
  - Load factor and resizing.
  - Hash collisions.
  - **HashMap** is **not synchronized** — use `ConcurrentHashMap` or `Collections.synchronizedMap()` if needed.
- 

Would you like code examples for **each method** with common Leetcode patterns like frequency maps, anagrams, or sliding windows?