Maps Overview

A Map is a data structure that stores key-value pairs.

- Keys are unique
- Values can be duplicated

Common Operations

operation	Description	Time complexity	
put(key, value)	add or update	O(1)	average
get(key)	retrieve a key for value	O(1)	
containsKey(key)	checks if the key exists	O(1)	
remove(key)	Remove key-value pair	O(1)	
size()	Number of entries	O(1)	

HashMap in Java

```
HashMap<String, Integer> map = new HashMap<>();
```

↳ doesn't maintain order

Map.Entry -> set of each pair of entries

- .getKey()
- .getValue()

```
map.computeIfAbsent(key, k -> new ArrayList<>()).add(str);
```

1- Check if the key exists

```
if (map.containsKey(key)) sout("exists");
```

2- Remove a key

```
map.remove(key);
```

3- Iterate over keys

```
for (String key: map.keySet()){
    sout (key + map.get(key));}
```

returns a set of all keys

4- Iterate over entries

```
for( Map.Entry<String, Integer> entry: map.entrySet()){
    sout (entry.getKey() + entry.getValue());}
```

getKey(): gets the key from this entry

getValue(): gets the value

entrySet(): returns the set of entries each containing a key & value

When to use in Leetcode problems?

a) Counting Frequencies

↳ counting how many times an element appears

b) Mapping values to indices

↳ store an element's index for O(1) lookup

c) Detecting duplicates

↳ check if the value has been seen