Lecture 48 | Hash Map

- `Hash-Map` : A data structure that stores <key, value> pairs.
- eg : `"Milind" : 4356, "Raj" : 1234, "Ravi" : 3456` (key-value pairs)
- Example problem : To find index of the key in array, arr = [1,2,3,4,5,6,7,8,9,10], key = 5 => index = 4 (index starts from 0)
- Lets say we need to find m elements in the array arr = [1,2,3,4,5,5,5,5,5,5] and key = 5, then we need to find the m elements in the array. => [5,5,5,5,5,5]
- Hashmap is an optimsation over normal traversal of array.
- Traversal of array is O(n) and hashmap/hashset is O(1)
- Searching in hashmap is O(1) (constant time) and insertion is O(1) (constant time) faster than array traversal.

Hashmap Operations

Operation	Time Complexity
Searching (value get...)	O(1) or O(Lambda)
Insertion (value put...)	O(1)
Check if present or not	O(1)

- Lambda : Hashing Constant

Hashmap / Hashtable (A deep dive into the topic)

- Hashmap is a data structure that stores key-value pairs. Lets take an example of a hashmap.

Country (key)	Population (value)
"India"	428
"China"	603
"USA"	400

- Hashmap combinations : <String, Integer>; <String, String>, <Integer, String>, <Integer, Integer>, <String, Double> ...
- Key is supposed to be unique.

Hashmap Operations in Java

`hm.put(key, value)` : Inserts the key-value pair into the hashmap. In O(1) time

> hm.put("India", 428); hm.put("China", 603); hm.put("USA", 400);

key	value

key	value
India	428
China	603
USA	400

> hm.put("India", 200); hm.put("USA", 28); hm.put("Dubai", 530);

key	value
India	200 (gets updated)
China	603
USA	28 (gets updated)
Dubai	530 (gets added)

- Important Points
  - When same key gets inserted, value gets updated.

`hm.get(key)` : Returns the value of the key. In O(1) time

> hm.get("India"); // Returns 200 hm.get("China"); // Returns 603 hm.get("USA"); // Returns 28 hm.get("Dubai"); // Returns 530 hm.get("Canada"); // Returns null

- Important Points
  - If key is not present, returns null.

`hm.containsKey(key)` : Returns true if key is present in the hashmap. In O(1) time

> hm.containsKey("India"); // Returns true hm.containsKey("China"); // Returns true hm.containsKey("USA"); // Returns true hm.containsKey("Dubai"); // Returns true hm.containsKey("Canada"); // Returns false

`hm.keySet()` : Returns the set of keys in the hashmap. In O(1) time

> hm.keySet(); // Returns Set {India, China, USA, Dubai}

Array vs Hashmap

- factorial storage

key	value
0	1
1	1

key	value
2	2
3	6
4	24
5	120

- to get the value of nth factorial, we need to store all the values in an array.

a[] = [1, 1, 2, 6, 24, 120]

- to get value of fact(5) = 120 can get O(1) time by using array and O(lambda) ~ O(1) time by using hashmap

Practical Example

```
import java.util.*;
// import java.util.HashMap;

public class Main {
    public static void main(String[] args) {
        // declare and initialise hashmap
        // HashMap<keyDataType, valueDataType> hashMapName = new
HashMap<keyDataType,
        // valueDataType>();
        HashMap<String, Integer> hm = new HashMap<String, Integer>();

        // add elements to hashmap
        hm.put("India", 628);
        hm.put("China", 837);
        hm.put("Dubai", 120);

        // print hashmap
        System.out.println(hm); // might not get in order due to hashmap
    }
}
```

> Output :

```
> java Main.java
{China=837, Dubai=120, India=628}
```

> On Updation :

```
import java.util.*;
// import java.util.HashMap;
```

```
public class Main {
    public static void main(String[] args) {
        // declare and initialise hashmap
        // HashMap<keyDataType, valueDataType> hashMapName = new
HashMap<keyDataType,
        // valueDataType>();
        HashMap<String, Integer> hm = new HashMap<String, Integer>();

        // add elements to hashmap
        hm.put("India", 628);
        hm.put("China", 837);
        hm.put("Dubai", 120);
        hm.put("India", 200);
        hm.put("Pak", 837);
        hm.put("USA", 443);

        // print hashmap
        System.out.println(hm); // might not get in order due to hashmap
    }
}
```

> Output :

```
> java Main.java
{USA=443, China=837, Pak=837, Dubai=120, India=200}
```

`hm.get(key)` : Returns the value of the key

```
// get value from hashmap
System.out.println(hm.get("India")); // Returns 200
```

`hm.containsKey(key)` : Returns true if key is present in the hashmap

```
// containsKey() method returns true if key is present in hashmap
boolean isChinaPresent = hm.containsKey("China"); // returns true or false
depending on key's presence
boolean isBangladeshPresent = hm.containsKey("Bangladesh"); // returns
false
System.out.println(isChinaPresent); // true
System.out.println(isBangladeshPresent); // false
```

`hm.keySet()` : Returns the set of keys in the hashmap

```java
// keySet() method returns set of keys
for (String key : hm.keySet()) {
    System.out.println(key);
}
```

`hm.size()` : Returns the number of key-value pairs in the hashmap

```java
// size
System.out.println(hm.size()); // 5 : number of elements/entry in hashmap
```