HASHMAP

Java HashMap class implements the Map interface which allows us to store key and value pair, where keys should be unique. If you try to insert the duplicate key, it will replace the element of the corresponding key. It is easy to perform operations using the key index like updation, deletion, etc.

Points to remember

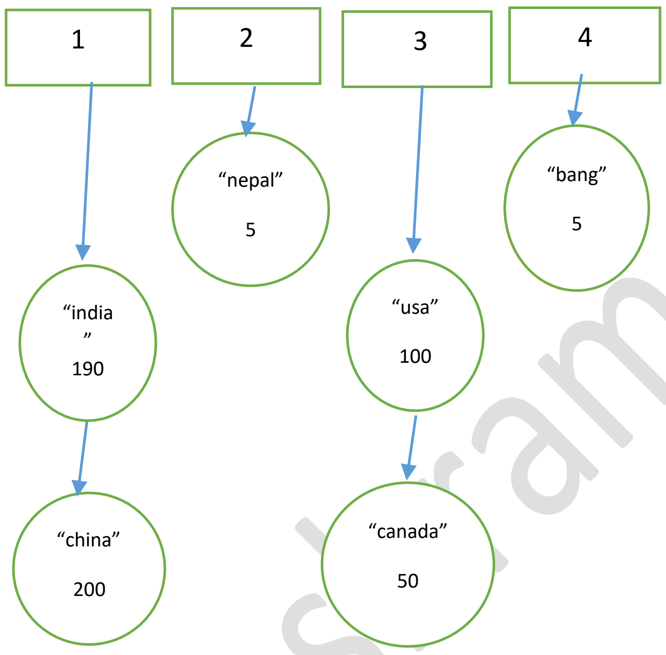
- Java HashMap contains values based on the key.
- Java HashMap contains only unique keys.
- Java HashMap may have one null key and multiple null values.
- Java HashMap is non synchronized.
- Java HashMap maintains no order.
- The initial default capacity of Java HashMap class is 16 with a load factor of 0.75.

⇨ Hashmap internally implements as an array of LL
⇨ Its time complexity is O(1)

Important Function:

1. put();
2. get();
3. containsKey();
4. remove();
5. size();
6. keyset();

key	value
“india”	190
“china”	200
“usa”	100
“nepal”	5
“Canada”	50
“Bangladesh”	5



```java
 package HashMap;

import java.util.HashMap;
import java.util.Map;
import java.util.*;

public class Introduction {
    public static void main(String[] args) {
        HashMap<String, Integer>  map = new HashMap<>();

        // insertion
        map.put("India", 120);
        map.put("China", 30);
        map.put("US", 10);
        System.out.println(map);

        // two cases it key is available then value is updated
        // else crete a new value
        map.put("China", 180);
        System.out.println(map);



        // search
        if (map.containsKey("China")) {
            System.out.println("present");
        }
        else {
            System.out.println("not present in map");
        }
        // get function
        System.out.println(map.get("China"));  // key exist
        System.out.println(map.get("Malaysia")); // key does not exist



        //
        int[] arr = {11, 12, 13, 14, 15, 16};

        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();


        for (int val: arr) {
            System.out.print(val + " ");
        }
        System.out.println();


        // for loop of hashmap
        // 1st way
        for (Map.Entry<String, Integer> e : map.entrySet()) {
            System.out.println(e.getKey() + " = " + e.getValue());
        }
        System.out.println();

        // 2nd way
        Set<String> key = map.keySet();
        for (String keys : key) {
            System.out.println(keys + " = " + map.get(keys));
        }


        // remove pair
        map.remove("China");
        System.out.println(map);

    }
}
```

Questions

// given an integer array of size n, find all elements that appear more than (n/3) times

```java
package HashMap;

import java.util.HashMap;

// given an integer array of size n, find all elements that appear more
than (n/3) times
public class MajorityElement {
    public static void main(String[] args) {
        int[] arr = {1, 3, 2, 5, 1, 3, 1, 5, 1};
        MajorityElement e = new MajorityElement();

        HashMap<Integer, Integer> map = new HashMap<>();
        for (int i = 0; i < arr.length; i++) {
            e.majority(map, arr[i]);
        }
        System.out.println(map);
        e.countTime(map, arr.length);

    }


    public void majority(HashMap<Integer, Integer> map, int element) {
        int count = 1;
        if (map.containsKey(element)) {
            map.put(element, map.get(element) + 1);
        }
        else {
            map.put(element, 1);
        }
    }


    public void countTime(HashMap<Integer, Integer> map, int n) {
        for (int key : map.keySet()) {
            if (map.get(key) > n / 3) {
                System.out.println(key);
            }
        }
    }
}
```

Union of 2 array

```java
package HashMap;

import java.util.HashSet;

public class UnionOfArray {
    public static void main(String[] args) {
        int[] arr1 =  {7, 3, 9};
        int[] arr2 = {6, 3, 9, 2, 9, 4};
        UnionOfArray u = new UnionOfArray();
        System.out.println( u.merge(arr1, arr2));
    }

    public int merge(int[] arr1, int[] arr2) {
        HashSet<Integer> set = new HashSet<>();

        for (int i = 0; i < arr1.length; i++) {
            set.add(arr1[i]);
        }
        for (int i = 0; i < arr2.length; i++) {
            set.add(arr2[i]);
        }

        System.out.println(set);

        System.out.print("the size is : " );
        return set.size();
    }

}
```

// intersection of 2 array

```java
package HashMap;

import java.util.HashSet;

// intersection of 2 arrays
public class Intersection_two_array {
    public static void main(String[] args) {
        int[] arr1 = {7, 3, 9};
        int[] arr2 = {6, 3, 9, 2, 9, 4};
        Intersection_two_array arr = new Intersection_two_array();
        System.out.println(arr.intersection(arr1, arr2));

    }


    public int intersection(int[] arr1, int[] arr2) {

        HashSet<Integer> set = new HashSet<>();
        for (int i = 0; i < arr1.length; i++) {
            set.add(arr1[i]);
        }

        int count = 0;
        for (int i = 0; i < arr2.length; i++) {
            if (set.contains(arr2[i])) {
                System.out.print(arr2[i] + " ");
                count++;
                set.remove(arr2[i]);
            }
        }
        System.out.println();

        System.out.print("Intersection : ");
        return count;
    }
}
```

Find itinerary of ticket

```
package HashMap;

import java.util.HashMap;

public class ItineraryTicket {
    public static void main(String[] args) {
        HashMap<String, String> ticket = new HashMap<>();
        ticket.put("Chennai", "Bengaluru");
        ticket.put("Mumbai", "Delhi");
        ticket.put("Goa", "Chennai");
        ticket.put("Delhi", "Goa");

        String start = getStart(ticket);

        while (ticket.containsKey(start)) {
            System.out.print(start + " -> ");
            start = ticket.get(start);
        }
        System.out.println(start);
    }


    public static String getStart(HashMap<String, String> ticket) {
        HashMap<String, String> revTick = new HashMap<>();

        for (String key : ticket.keySet()) {
            revTick.put(ticket.get(key), key );
        }

        for (String key: ticket.keySet()) {
            if (!revTick.containsKey(key)) {
                return key;
            }
        }
        return null;
    }

}
```

subarray sum equal to k

```java
package HashMap;

import java.util.HashMap;

// subarray sum equal to k
public class SubarraySum {
    public static void main(String[] args) {
        HashMap<Integer, Integer> map = new HashMap<>();
        int[] arr = {10, 2, -2, -20, 10};
        int k = -10;

        map.put(0, 1);    // because of empty map 0 exists once
        int sum = 0;
        int freq = 0;
        for (int i = 0; i < arr.length; i++) {
            sum += arr[i];

            if (map.containsKey(sum)) {
                map.put(sum, map.get(sum) + 1);
            }

            else  {
                map.put(sum, 1);
            }

            if (map.containsKey(sum - k)) {
                freq += map.get(sum - k);
            }
        }
        System.out.println(freq);
    }
}
```