employee management

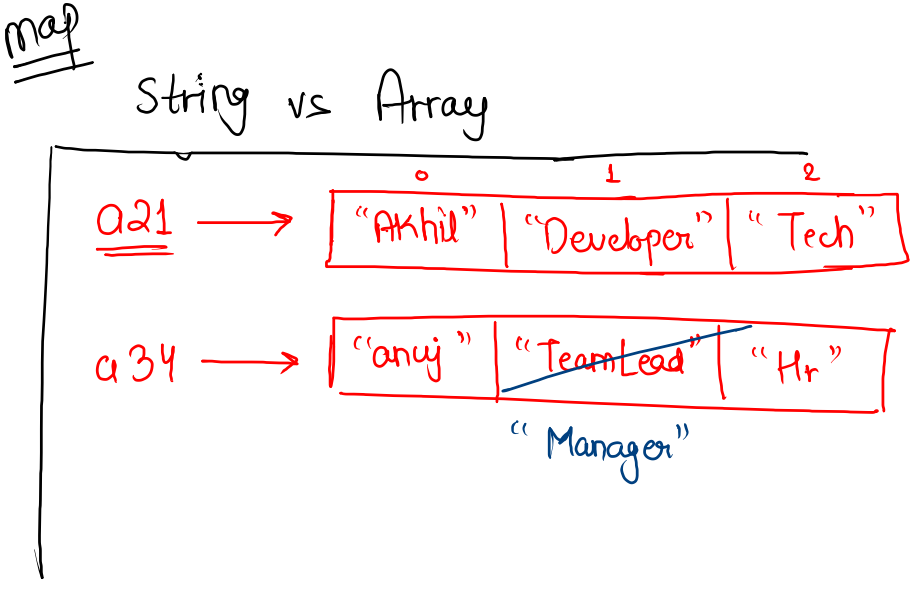
ip

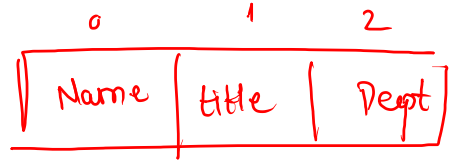
```
5
add a21 Akhil Developer Tech
add a34 anuj TeamLead Hr
update a34 Manager
delete a21
show a34
```

you will be getting **T** queries which includes:

1. case-1 (add) -> add employee with details.
2. case-2 (update) -> update job title of a given employee.
3. case-3 (delete) -> remove the employee.
4. case-4 (show) -> print details of a given employee else print -1.

map





Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    HashMap<String, ArrayList<String>> map = new HashMap<>();
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        String operation = scn.next();
        if ( operation.equals("add") ) {
            String empId = scn.next();
            String empName = scn.next();
            String empTitle = scn.next();
            String empDept = scn.next();

            ArrayList<String> arr = new ArrayList<>();
            arr.add(empName);
            arr.add(empTitle);
            arr.add(empDept);

            map.put( empId, arr );
        } else if ( operation.equals("update") ) {
            String empId = scn.next();
            String empTitle = scn.next();

            ArrayList<String> arr = map.get(empId);
            arr.set(1, empTitle); // index, updated_values

            map.put( empId, arr );

            // map.put( empId, map.get(empId).set(1, emptitle));
        } else if ( operation.equals("delete") ) {

            String empId = scn.next();
            map.remove(empId);

        } else if ( operation.equals("show") ) {
            String empId = scn.next();

            if ( map.containsKey(empId) == true ) {
                ArrayList<String> arr = map.get(empId);
                System.out.println( arr.get(0) + " " + arr.get(1) + " " + arr.get(2) );
            } else {
                System.out.println("-1");
            }
        }
    }
}
```

Unique Number of Occurrences

arr = [3, 5, 5, 7, 3, 3, 3]

map

number vs freq

3 → 4
5 → 2
7 → 1

map.values();

==

set

4
2
1

True

if size of map and size of set is same then true else false

→ Variation of hashmap

HashSet

// best used to indentify duplicacy because it only contain unique values

syntax

```
HashSet<Integer> set = new HashSet<>();
```

set

3
1
5
7
10

- ✓✓ set.add(key);
- set.remove(key);
- ✓✓ set.contains(key);
- set.size();

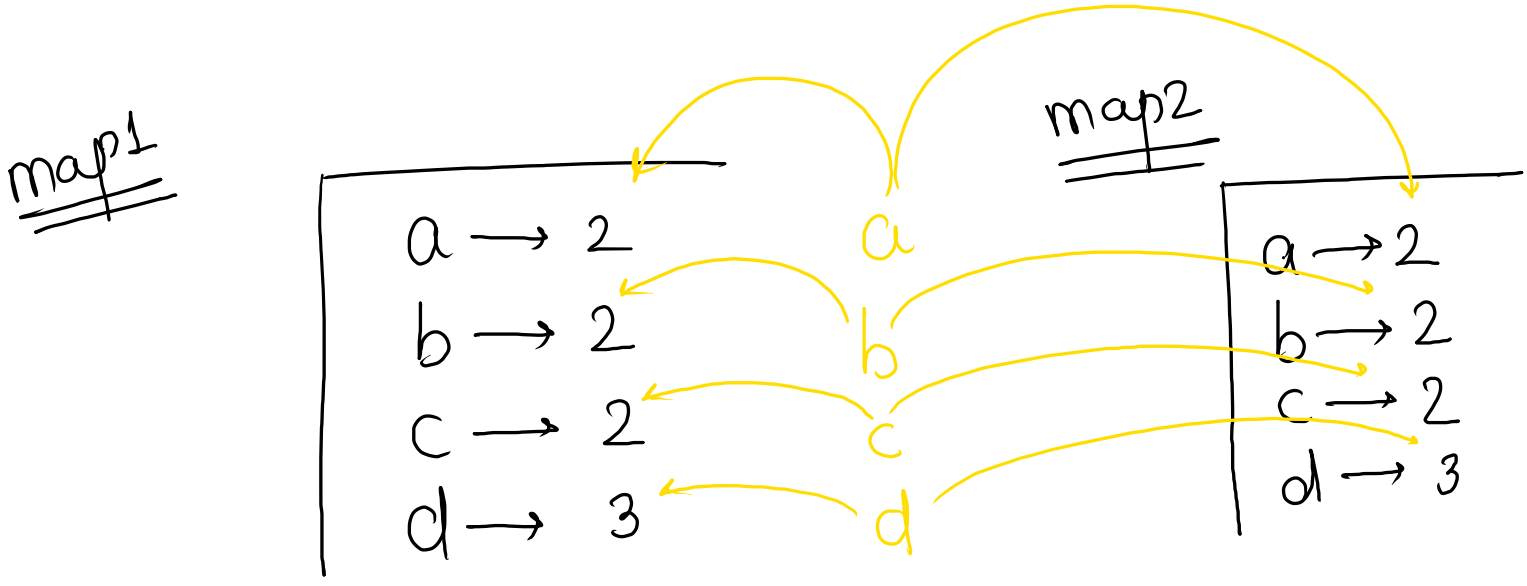
code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    System.out.println(uniqueOcc(arr, n));
}
public static boolean uniqueOcc(int[] arr, int n) {

    HashMap<Integer, Integer> map = new HashMap<>();
    for (int i = 0; i < n; i++) {
        if ( map.containsKey(arr[i]) == false ) {
            map.put( arr[i], 1 );
        } else {
            int freq = map.get(arr[i]);
            map.put( arr[i], freq + 1 );
        }
    }

    HashSet<Integer> set = new HashSet<>();
    for (int i : map.values()) {
        set.add(i);
    }

    if ( map.size() == set.size() ) {
        return true;
    } else {
        return false;
    }
}
```

Valid Anagram 5

str1 = "abcdabcdd"

str2 = "aabbccddd"



Code

O(N)

```java
public static boolean validAnagram(String str1, String str2) {
    HashMap<Character, Integer> map1 = new HashMap<>();
    for (int i = 0; i < str1.length(); i++) {
        char ch = str1.charAt(i);
        if ( map1.containsKey(ch) == false ) {
            map1.put( ch, 1 );
        } else {
            int freq = map1.get(ch);
            map1.put( ch, freq + 1 );
        }
    }
```

→ N

O(N)

```java
    HashMap<Character, Integer> map2 = new HashMap<>();
    for (int i = 0; i < str2.length(); i++) {
        char ch = str2.charAt(i);
        if ( map2.containsKey(ch) == false ) {
            map2.put( ch, 1 );
        } else {
            int freq = map2.get(ch);
            map2.put( ch, freq + 1 );
        }
    }
```

→ N

O(N)

```java
    for (Map.Entry<Character, Integer> e : map1.entrySet()) {
        char ch = e.getKey();
        int freq = e.getValue();
        // check character should be same
        if ( map2.containsKey(ch) == false ) {
            return false;
        }
        // and its freq should be same as well
        if ( map2.get(ch) != map1.get(ch) ) {
            return false;
        }
    }
    return true;
}
```

T.C = O(N)

S.C = O(N)