https://course.acciojob.com/idle?question=86402bd0-eeed-4c05-bf51-6ef08065b6c8

- EASY
- Max Score: 30 Points

Design HashSet

Design a HashSet without using any built-in hash table libraries.

You are required to complete the following functions:

`void add(key) Inserts the value key into the HashSet.`

`bool contains(key) Returns whether the value key exists in the HashSet or not.`

`void remove(key) Removes the value key in the HashSet. If key does not exist in the HashSet, do nothing.`

Input Format

You are given `Q` **number of queries which have values** `1-3`**.**

For each query you are also given a key.

Input means that following

1-> add() function is called with a key as parameter

2-> contains() function is called with a key as parameter.

3-> remove() function is called with key as parameter.

Output Format

Output is managed for you.

Example 1

Input

```
8
1 1
1 2
2 1
2 3
1 2
2 2
3 2
2 2
```

Output

```
True False True False
```

Explanation

```
1 1 -> Obj.add(1);      // set = [1]
1 2 -> Obj.add(2);      // set = [1, 2]
2 1 -> Obj.contains(1); // return True
2 3 -> Obj.contains(3); // return False, (not found)
1 2 -> Obj.add(2);      // set = [1, 2]
2 2 -> Obj.contains(2); // return True
3 2 -> Obj.remove(2);   // set = [1]
2 2 -> Obj.contains(2); // return False, (already removed)
```

Example 2

Input

```
4
1 2
1 3
2 2
2 4
```

Output

```
True False
```

Explanation

```
1 2 -> Obj.add(2);      // set = [2]
```

```
1 3 -> Obj.add(3);      // set = [2, 3]
2 2 -> Obj.contains(2); // return True
2 4 -> Obj.contains(3); // return False, (not found)
```

Constraints

0 <= key <= 10^6

At most 10^4 calls will be made to `add`, `remove`, and `contains`.

Topic Tags

- Hashing

My code

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

class Solution {
        HashSet<Integer> hs;
        public Solution()
        {
        hs = new HashSet<Integer>();
        }
    public void add(int key) {
        //Complete the function
                hs.add(key);
    }

    public void remove(int key) {
        //Complete the function
                hs.remove(key);
```

```
    }

    public boolean contains(int key) {
        //Complete the function
                if(hs.contains(key))
                        return true;
                return false;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int q = sc.nextInt();
        Solution Obj = new Solution();
        for (int i = 0; i < q; i++) {
            int choice = sc.nextInt();
            int key = sc.nextInt();
            if (choice == 1) {
                // Add
                Obj.add(key);
            } else if (choice == 2) {
                // contains
                if (Obj.contains(key)) {
                    System.out.print("True ");
                } else {
                    System.out.print("False ");
                }
            } else if (choice == 3) {
                // remove
                Obj.remove(key);
            }
```

```
        }
        System.out.println();
        sc.close();
    }
}
```