practice Question

find freq. of each element

1	2	-1	2	-1	1	1

```
if (!map.containKey(arr[i])) {
    map.put(arr[i], 1);
} else {
  map.put(arr[i],
          map.get(arr[i])+1)
}
```

Integer, Integer

1 → ~~1~~ ~~2~~ 3

2 → ~~1~~ 2

-1 → ~~1~~ 2

Ex2

arr

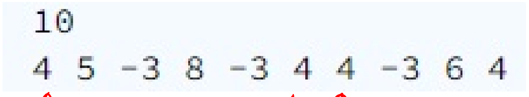
0	1	2	3	4	5	6	7
-1	2	5	6	8	5	2	-1

```
for(int i=0; i<arr.length; i++){
    if( map.containsKey(arr[i]) == false){
        map.put( arr[i], 1);
    } else {
                                       -1
    map.put(arr[i], map.get(arr[i])+1);
            -1            1
                          2
    }
}
```

HM

-1 → 2
2 → 2
5 → 2
6 → 1
8 → 1

Same Number Same Frequency



```
10
4 5 -3 8 -3 4 4 -3 6 4
```

map

4 → 4

5 → 1

-3 → 3

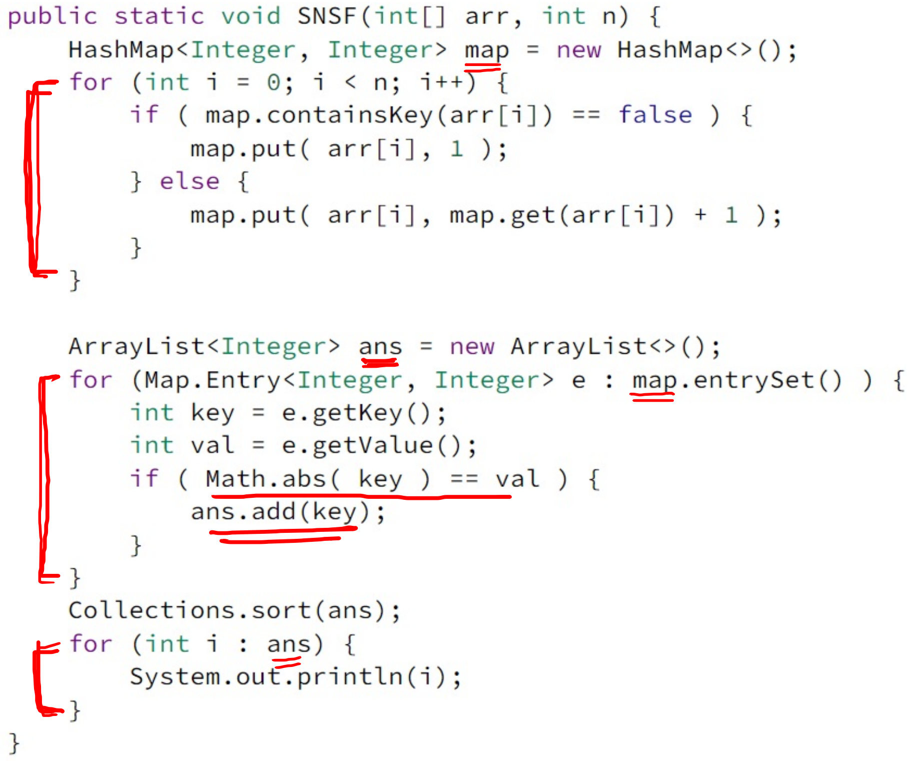
8 → 1

6 → 1

4, -3

|-3| == 3

traverse in hashmap



```java
public static void SNSF(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 {
            map.put( arr[i], map.get(arr[i]) + 1 );
        }
    }

    ArrayList<Integer> ans = new ArrayList<>();
    for (Map.Entry<Integer, Integer> e : map.entrySet() ) {
        int key = e.getKey();
        int val = e.getValue();
        if ( Math.abs( key ) == val ) {
            ans.add(key);
        }
    }
    Collections.sort(ans);
    for (int i : ans) {
        System.out.println(i);
    }
}
```

Character and it's Frequency

[a , - , * ,   , _ , Z]

'*'

map   Character, Integer

a → 1

- → 1

* → ~~1~~ 2

  → 1

_ → 1

Z → 1

```java
public static void charFreq(char[] arr, int n) {
    HashMap<Character, Integer> map = new HashMap<>();
    for (int i = 0; i < n; i++) {
        if ( map.containsKey(arr[i]) == false ) {
            map.put( arr[i], 1 );
        } else {
            map.put( arr[i], map.get(arr[i]) + 1 );
        }
    }

    ArrayList<Character> ans = new ArrayList<>();
    for (Map.Entry<Character, Integer> e : map.entrySet()) {
        char key = e.getKey();
        int val = e.getValue();
        ans.add(key);
    }
    Collections.sort(ans);
    for (char c : ans) {
        System.out.println( c + " " + map.get(c) );
    }
}
```

S.C = O(N)

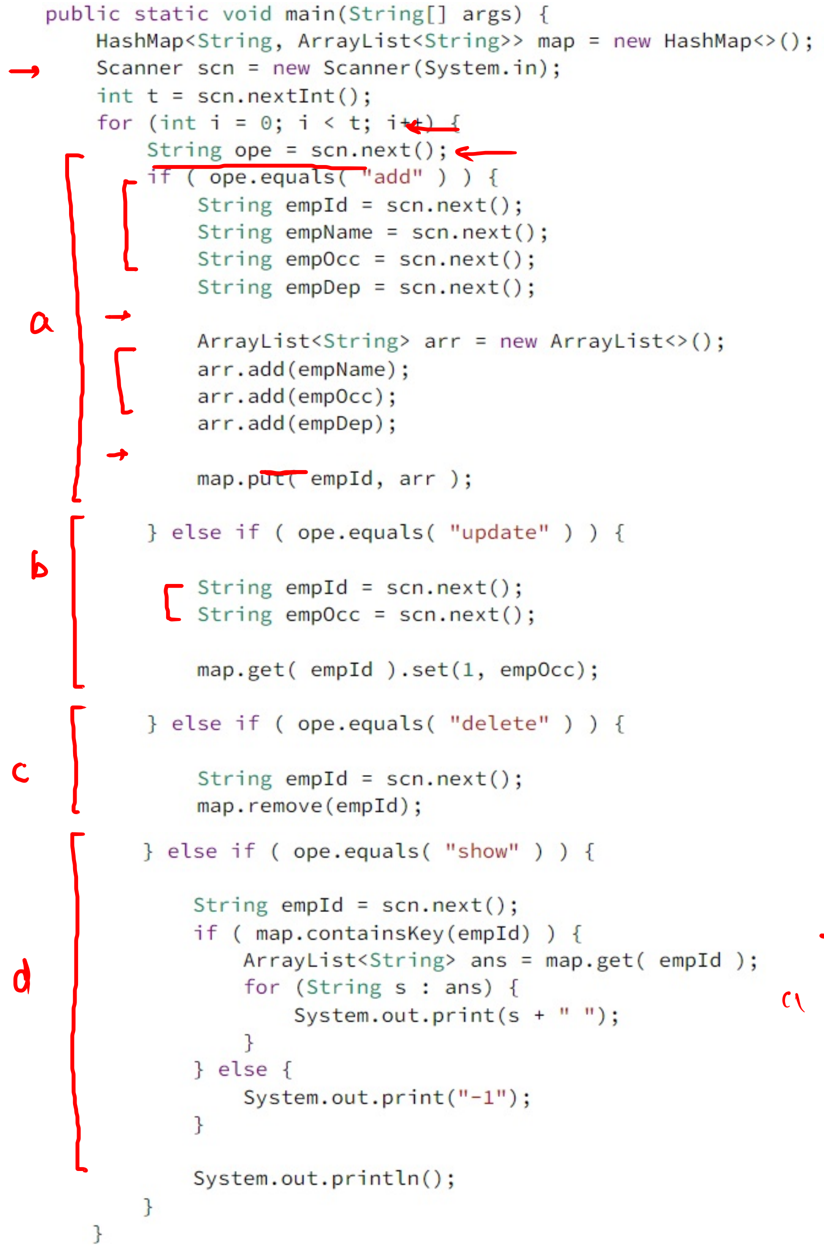
T.C $\boxed{O(N)}$

Note:-

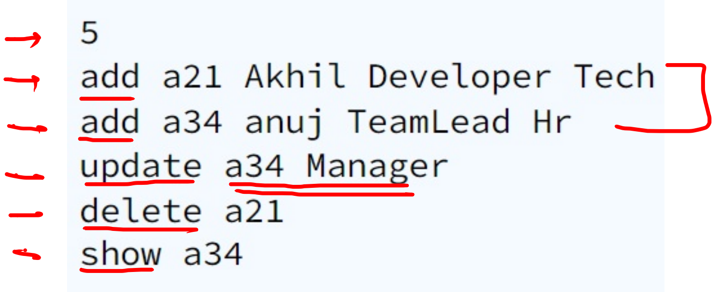
amotised O(1)

operation = 1

$\cong$ 3 - 4



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

            ArrayList<String> arr = new ArrayList<>();
            arr.add(empName);
            arr.add(empOcc);
            arr.add(empDep);

            map.put( empId, arr );

        } else if ( ope.equals( "update" ) ) {

            String empId = scn.next();
            String empOcc = scn.next();

            map.get( empId ).set(1, empOcc);

        } else if ( ope.equals( "delete" ) ) {

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

        } else if ( ope.equals( "show" ) ) {

            String empId = scn.next();
            if ( map.containsKey(empId) ) {
                ArrayList<String> ans = map.get( empId );
                for (String s : ans) {
                    System.out.print(s + " ");
                }
            } else {
                System.out.print("-1");
            }

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



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

t = 5

operation = "add"

empId = "a21"

empName = "Akhil"

empOcc = "Developer"

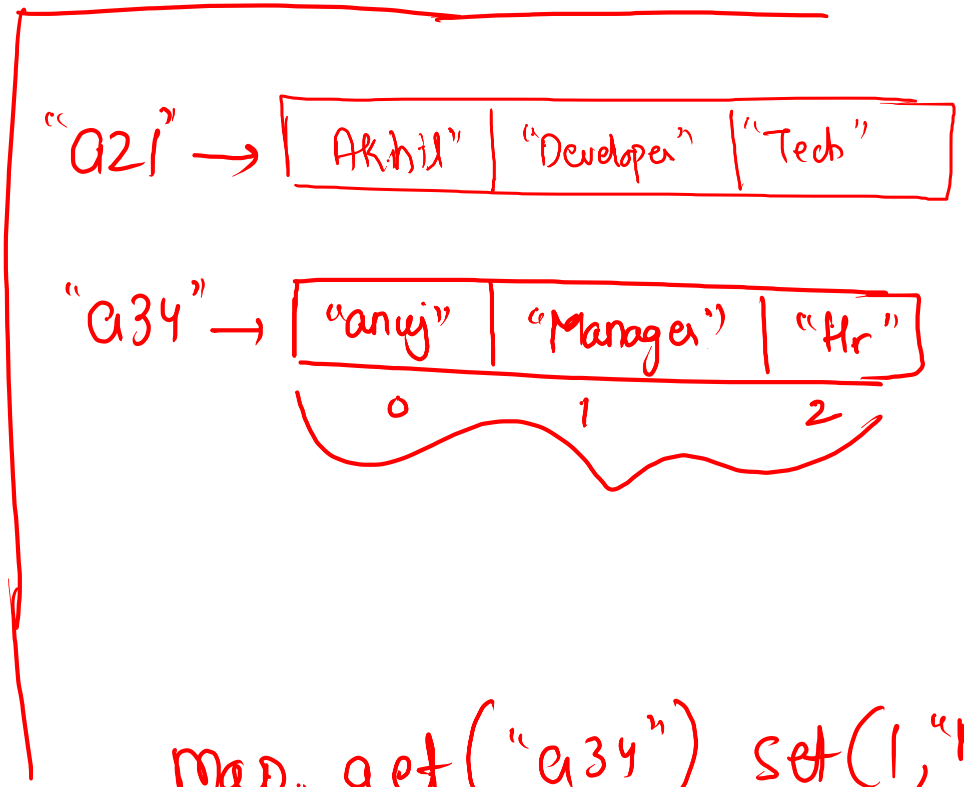
empDep = "Tech"

"

empId = "a34"

empOcc = "Manager"

String, AL



```
map.get("a34").set(1,"Manager");
```

Two Sum 14

$$x_1 + x_2 == target$$

arr	2	1	3	6	4	7

$$x_2 == target - x_1$$

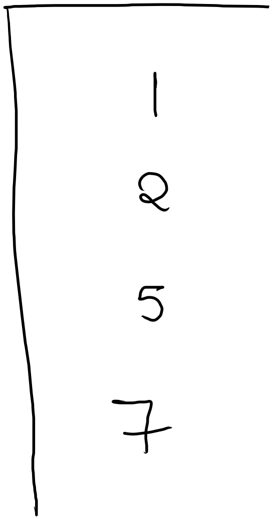
target = 9

ans 2, 7

3, 6

Hashset < Integer >

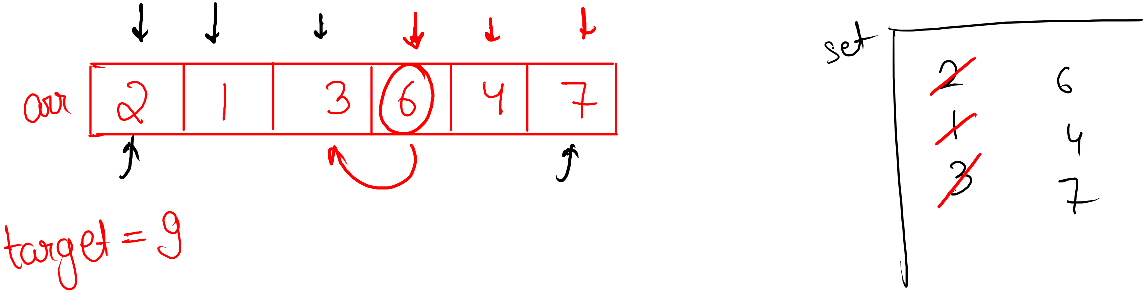
set



set.add(val)

set.remove(val)

set.contains(val)



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

for ( ______________________ ) {
        set.add(arr[i]);
}
```

target = 9

→ $x_1 = 2, \quad x_2 = 7$

$x_1 = 1, \quad x_2 = 8$

→ $x_1 = 3, \quad x_2 = 6$

$x_1 = 6, \quad x_2 = 3$

$x_1 = 4, \quad x_2 = 5$

$x_1 = 7, \quad x_2 = 2$

$x_1 == i$

$x_2 != i$

val → idx

2 → 0

1 → 1

3 → 2

6 → 3

4 → 4

7 → 5