Practice

```
public class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("abc", 2);
        map.put("efg", 7);
        map.put("xyz", 5);
        map.put("efg", 8);
        System.out.println(map);
        map.remove("efg");
        System.out.println(map);
        System.out.println(map.get("abc"));
        System.out.println(map.get("efg"));
        System.out.println( map.containsKey("efg") );
        System.out.println( map.containsValue(2) );
        System.out.println( map.values() );
    }
}
```

Same Number Same Frequency

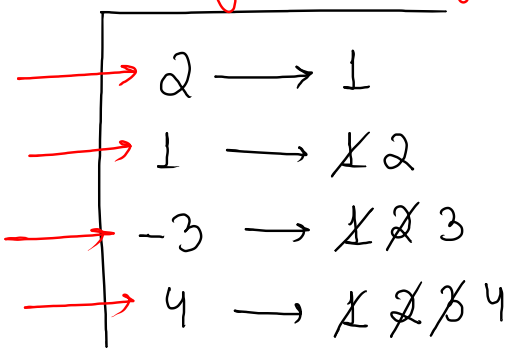
n = 10

arr = [2, 1, -3, 4, 1, 4, -3, 4, 4, -3]

(number) (freq)

hashmap

Integer vs Integer



Math.abs(key) == value

|2| == 1
|1| == 2
|-3| == 3 ✓
|4| == 4 ✓

1) loop from 0 to n-1

1.1) if curr val is not present in hashmap

map.put(curr, 1)

1.2) else

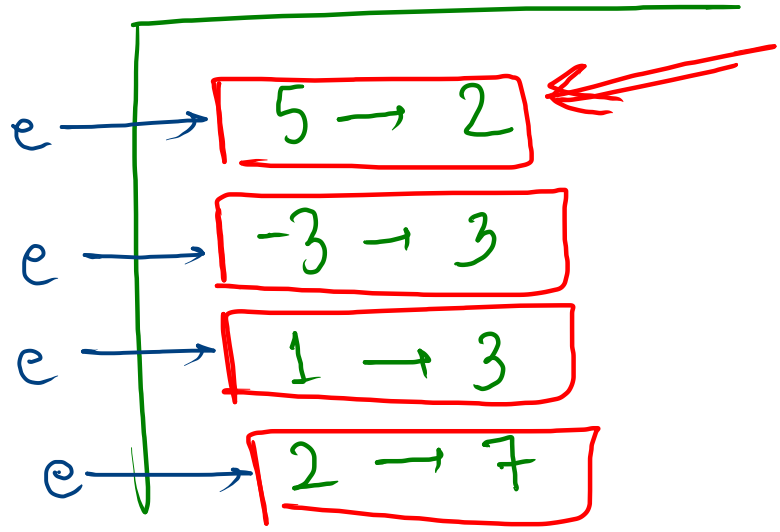
int freq = map.get(curr);

map.put(curr, freq+1);

→ only way to traverse in HM

name of HM

```
for (Map.Entry<Integer, Integer> e : map.entrySet()){
    Syso ( e.getKey() );
    Syso ( e.getValue() );
}
```



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

    ArrayList<Integer> ans = new ArrayList<>();
    for (Map.Entry<Integer, Integer> e : map.entrySet()) {
        int key = e.getKey();
        int value = e.getValue();
        if ( Math.abs(key) == value ) {
            ans.add(key);
        }
    }

    Collections.sort(ans);
    for (int i : ans) {
        System.out.println(i);
    }
}
```

O(N)

Character and it's Frequency

Character vs Integer

map

a → 2
c → 2
b → 2
z → 1

any

a – 2
b → 2
c → 2
z → 1

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    char[] arr = new char[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.next().charAt(0);
    }
    charByFreq(arr, n);
}
public static void charByFreq(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 value = e.getValue();
        ans.add(key);
    }

    Collections.sort(ans);
    for (char i : ans) {
        System.out.println( i + " " + map.get(i) );
    }
}
```

Two Sum 14

n = 4

0	1	2	3
2	7	11	15

arr =

target = 9 (num1 + num2 == tar)

Ans = 0 , 1

num1 = 2 , num2 = tar - num1

= 7

map

2 → 0

7 → 1

11 → 2

15 → 3

Integer vs Integer

(number vs index)

map.get(num1) , map.get(num2)

arr $= [-2, 4, 6, 0, 3, 1]$ (indices 0, 1, 2, 3, 4, 5)

target = 3

number → index

number → index
$-2 \rightarrow 0$
$4 \rightarrow 1$
$6 \rightarrow 2$
$0 \rightarrow 3$
$3 \rightarrow 4$
$1 \rightarrow 5$

num1 = −2, num2 = tar − num1 = 5

num1 = 4, num2 = −1

num1 = 6, num2 = −3

num1 = 0, num2 = 3

map.get(num1), map.get(num2)

(3) (4)

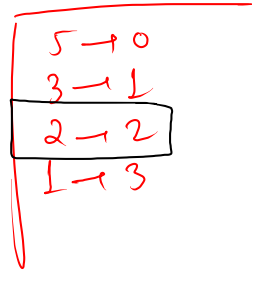
Code

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

    twoSum(arr, n, k);
}
public static void twoSum(int[] arr, int n, int k) {
    HashMap<Integer, Integer> map = new HashMap<>();
    for (int i = 0; i < n; i++) {
        map.put( arr[i], i );
    }

    for (int i = 0; i < n; i++) {
        int num1 = arr[i];
        int num2 = k - num1;
        if ( map.containsKey(num2) == true ) {
            if ( i != map.get(num2) ) {
                int[] ans = {i, map.get(num2)};
                Arrays.sort(ans);
                System.out.println( ans[0] + " " + ans[1] );
                return;
            }
        }
    }
}
```

i
map.get(num2)

arr = [5 , 3 , 2 , 7]
0 1 2 3

target = 4



num1	num2
5	-1
3	1
2	2

T.C = O(N)