

HashMap →

Usage + Application

Key-value pair

{ }

[]

Operations

- put ()
- get ()
- containsKey ()
- Size ()
- keySet ()
- remove ()

Array → Index → get
HashMap → Key → put

Key value
<String , Integer>

India	<u>200</u>
UK	100
USA	200
PAK	100

Element already exists

True

False

$O(1)$ put(k,v)

Update

Insert

$O(1)$ get(k)

returns value

return null

$O(1)$ remove(k)

removes pair &
returns value

return null

$O(1)$ containsKey(k)

returns true

returns false

$O(1)$ size() →

number of pairs in hashmap.

$O(n)$ keySet() →

India → 200

USA → 300

China → 250

UK → 195

→ Loop X

(keySet)

India | USA | china | UK

Loop

zmszeqxllyheqwrfgcunypejcxovtaqbnqyqlmrwite
 ↑↑↑↑↑↑↑↑↑↑

Character Integer

z 1x3
 m 1
 s 1
 e 1x2
 q 1x2
 x 1
 l 1x2
 v 1
 h 1

Print

5 min

max freq char

freq Map

```
String inp = scn.nextLine();
HashMap<Character, Integer> hm = new HashMap<>();

char hfCh = ' ';
int hFreq = 0;
for(int i = 0 ; i < inp.length() ; i++){
    char ch = inp.charAt(i);

    if(hm.containsKey(ch)){
        hm.put(ch, hm.get(ch)+1 );
    }else{
        hm.put(ch,1);
    }

    if(hm.get(ch) > hFreq){
        hFreq = hm.get(ch);
        hfCh = ch;
    }
}

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

inp = a a b c a b a c b b c b b b a
 ↑↑↑↑↑↑↑↑↑↑↑↑↑↑

a 1x2x3x4
 b 1x2x3x5
 c 1x2x3

hfCh = ~~b~~
 hFreq = 5

if(ch < hfCh) {

}

arr1 =

0	1	2	3	4	5	6	7	8
5	5	9	8	5	5	8	0	3

arr1 \rightarrow key map

arr2 =

9	7	1	0	3	6	5	9	1	1	8	0	2	4	2	9	1	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

7-8

key

value

5	3
9	1
8	2
0	1
3	1

$$\begin{bmatrix} 9 \\ 0 \\ 3 \\ 5 \\ 8 \end{bmatrix}$$

⑦

1 1 2 2 2 3 5

⑦

1 1 1 2 2 4 5

$a_1 =$

0	1	2	3	4	5	6
1	1	2	2	2	3	5

$a_2 =$

0	1	2	3	4	5	6
1	1	1	2	2	4	5
↑	↑	↑	↑	↑	↑	↑

1
1
2
2
5

1	0 1 2
2	1 2 3
3	1
5	0 1

$a_i \rightarrow \text{freq map}$

$\text{fl. w.} =$

Integer

num

17 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ . . .
12 5 1 2 10 2 13 7 11 8 9 11 8 9 5 6 11

2

T_{2m}

12	Tone
5	Tone
1	Tone
2	Tone
10	Tone
13	Tone
7	Tone
11	Tone
8	Tone
9	Tone
6	Tone

→	$\frac{12}{5}$ ✓	Falsch
→	$\frac{5}{10}$ ✓	Richtig
→	1 ✓	Richtig
→	2 ✓	Falsch
→	10 ✓	Falsch
→	13 ✓	Falsch
→	7 ✓	Falsch
→	11 ✓	Falsch
→	8 ✓	Falsch
→	9 ✓	Falsch
→	6 ✓	Falsch

✓ $0Sp = 15$

✓ counter = 89

$$S_p = 1$$

Counter = 12

```
int oSp = -1 , oCounter = 0;

for(int key : hm.keySet()){
    if(hm.get(key)){
        int sp = key , counter = 1;
        while(hm.containsKey(sp+counter)){
            counter++;

            if(counter > oCounter){
                oSp = sp;
                oCounter = counter;
            }
        }
    }
}

for(int i = 0 ; i < oCounter ; i++){
    System.out.println(oSp+i);
}
```

$\Rightarrow \underline{\underline{2.7}}$

$$\underline{\underline{n}}$$

0	1	2	3	4	5	6	7	8
↓	↓	↓	↓	↓	↓	↓	↓	↓
5	6	7	8	9	10	11	12	13

```
HashMap<Integer, Boolean> hm = new HashMap<>();
```

```
for(int v1 : arr){
    hm.put(v1,true);
}
```

```
for(int key : hm.keySet()){
    if(hm.containsKey(key-1)){
        hm.put(key, false);
    }
}
```

```
int oSp = -1 , oCounter = 0;
```

```
for(int key : hm.keySet()){
    if(hm.get(key)){
        int sp = key , counter = 1;
        while(hm.containsKey(sp+counter)){
            counter++;
        }

        if(counter > oCounter){
            oSp = sp;
            oCounter = counter;
        }
    }
}
```

```
for(int i = 0 ; i < oCounter ; i++){
    System.out.println(oSp+i);
}
```

```

for(int v1 : arr){
    hm.put(v1,true);
}

for(int key : hm.keySet()){
    if(hm.containsKey(key-1)){
        hm.put(key,false);
    }
}

int oSp = -1 , oCounter = 0;

for(int key : hm.keySet()){
    if(hm.get(key)){
        int sp = key , counter = 1;
        while(hm.containsKey(sp+counter)){
            counter++;
        }

        if(counter > oCounter){
            oSp = sp;
            oCounter = counter;
        }
    }
}

for(int i = 0 ; i < oCounter ; i++){
    System.out.println(oSp+i);
}

```

$\Rightarrow 2n$

$\Rightarrow n$

\checkmark \checkmark $(Sn) \rightarrow C.n \rightarrow \underline{O(n)}$

\checkmark \checkmark $S.C. \rightarrow \underline{O(n)}$