HashMap → ✪ Usage + Application

→ Key-value pair

→ Operations
- put( )
- get( )
- ContainsKey()
- Size()
- Keyset()
- remove()

{ }

[ ]

Array → Index → get
put

Hashmap → Key

Key value
<String , Integer>

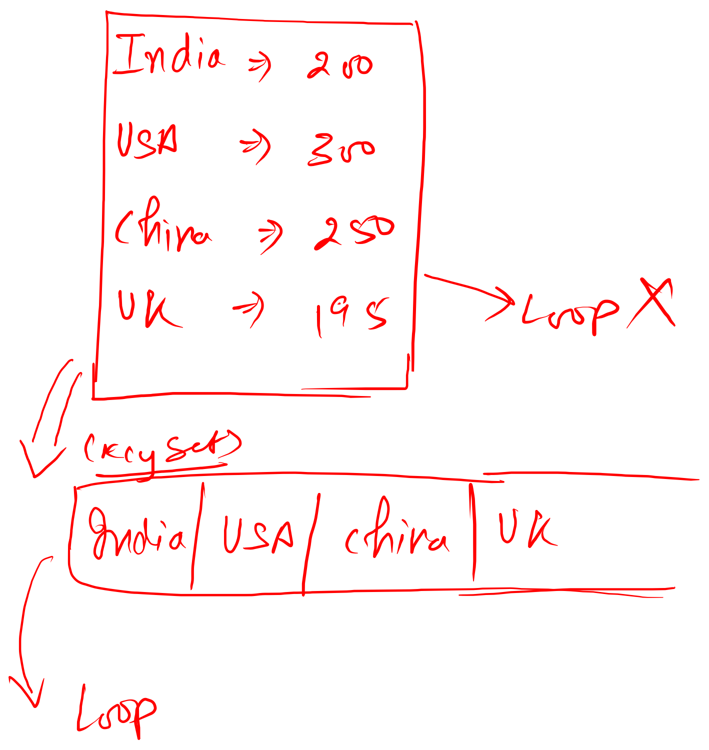
India	200
UK	180
USA	290
PAK	180

Element already Exists

		True	False
O(1)	put(K,V)	Updation	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() →



zmszeqxllzvheqwrofgcuntypejcxovtaqbnqyqlmrwitc

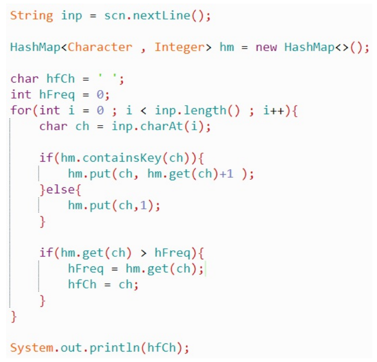
Character	Integer
z	~~1~~ ~~2~~ 3
m	1
s	1
e	~~1~~ 2
q	~~1~~ 2
x	1
l	~~1~~ 2
v	1
h	1

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	~~1~~ ~~2~~ ~~3~~ 4
b	~~1~~ ~~2~~ ~~3~~ ~~4~~ 5
c	~~1~~ ~~2~~ 3

hfCh = ~~a~~ b

hFreq = ~~4~~ 5

if(ch < hfCh){

arr1 → HM
↳ freq map

arr1 =

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

7-8

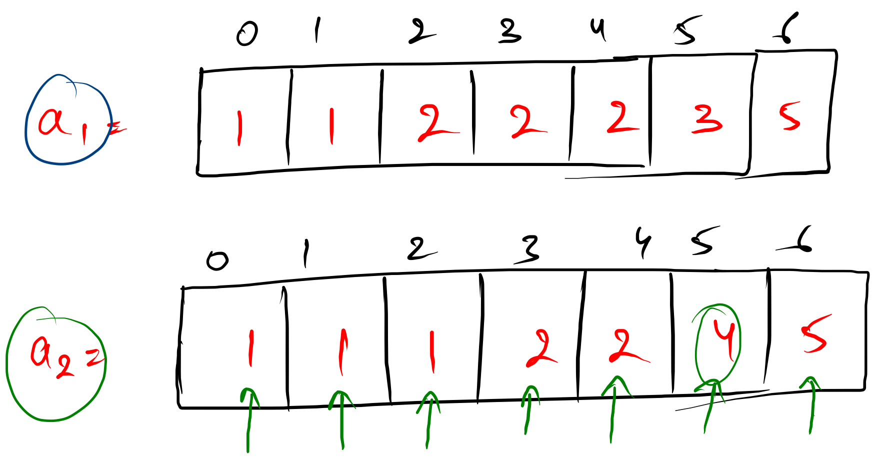
arr2 =

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

<Integer	Integer>
~~5~~	~~3~~
~~9~~	~~1~~
~~8~~	~~2~~
~~0~~	~~1~~
~~3~~	~~1~~

[9, 0, 3, 5, 8]

7
1 1 2 2 2 3 5
7
1 1 1 2 2 4 5



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

$a_1 \rightarrow$ freq map

H.W. ✓

1
1
2
2
5

Integer

null

17

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

<Integer , Boolean>

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

$\boxed{2}$

True

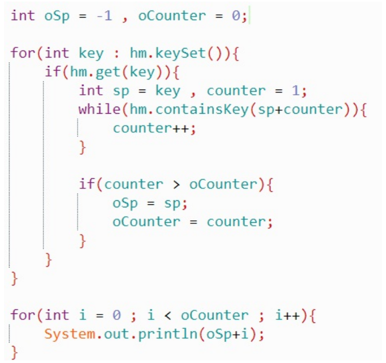
12 ✓	False
5 ✓	True
1 ✓	True
2 ✓	False
10 ✓	False
13 ✓	False
7 ✓	False
11 ✓	False
8 ✓	False
9 ✓	False
6 ✓	False

✓ oSp = ~~-1~~ 5

✓ oCounter = ~~0~~ 9

Sp = 1

Counter = ~~1~~ 2

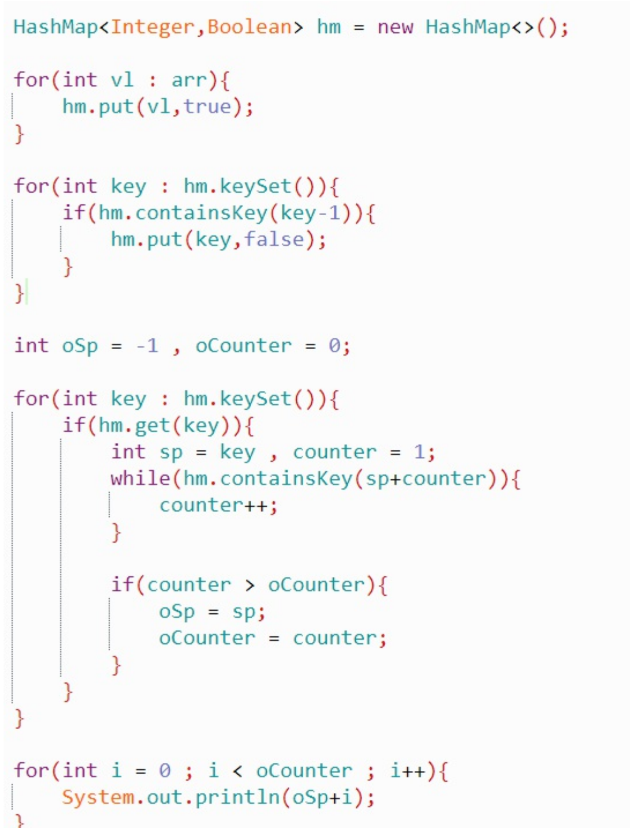


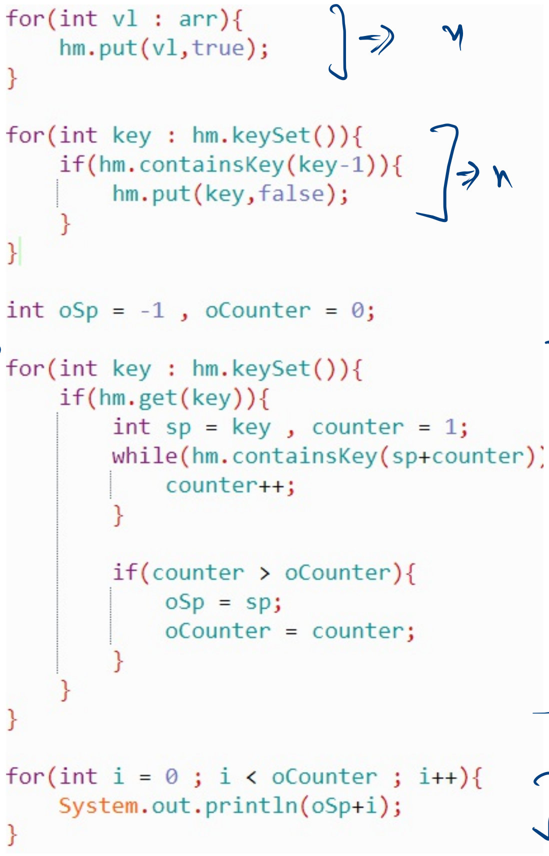
```
int oSp = -1 , oCounter = 0;

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);
}
```

$\Rightarrow \boxed{2n}$

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 vl : arr){
    hm.put(vl,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 vl : arr){
    hm.put(vl,true);
}
```

$\Rightarrow n$

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

$\Rightarrow n$

```
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;
        }
    }
}
```

$\Rightarrow 2n$

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

$\Rightarrow n$

$\boxed{5n} \Rightarrow$ T.C $\Rightarrow O(n)$

S.C. $\Rightarrow O(n)$