HashSet
{
==> stores unique elements
==> to check for a particular element in a faster manner

[10, 20, 30, 50, 60, 100]

java please check whether element 100 is avaiable OR not

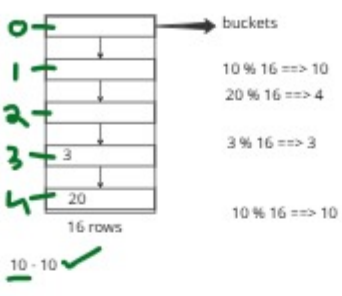
search for a particualr element ==> HashSet

{
1000 elements ==> 40 ==> 1s
10000 elements ==> 40 ==> 1s
100000 elements ==> 40 ==> 1s
}

HasSet ==> [10, 20, 30, 40, 70, 3, 100]

element = 10

HashSet internally stores the data inside a HashTable



10 - 10

```
if(!dataSet.contains(3))
{
  add
}
```

MOVE ZEROES

[0,1,0,3,12]

[1, 3, 12, 0, 0]

[0,1,0,3,12]

iterate each

zero element ==> dont do
non zero element ==> opetration

[1, 1, 0, 3, 12]

[1, 3, 0, 3, 12]

[1, 3, 12, 3, 12]

[1, 3, 12, 0, 0]

Whenever we encotuter first non zero element
==> kept at the 0th index
Whenever we encotuter second non zero element ==> kept at the 1th index

.....

[0, 0, 4, 0, 5, 6, 9, 0, 2]

[4, 5, 6, 9, 2, 0, 0, 0, 0]

[100, 4, 200, 1, 3, 2]

[100, 4, 55, 101, 102, 1, 3, 2, 103, 10, 11]

1 => 2 => 3 => 4
10 => 11
100 => 101 => 102 => 103
55

[100, 4, 200, 1, 3, 99, 101, 2]

iterate through each element

i = 100 - 1 ==> 99

99 - 1 ==> 98

["eat","tea","tan","ate","nat","bat"]

```
[
  ["bat"],
  ["nat","tan"],
  ["ate","eat","tea"]
]
```

["eat","tea","tan","ate","nat","bat"]

"eat" ==> ['e', 'a', 't'] ==> sorting ==> ['a', 'e', 't'] ==> string ==> "aet"
"tea" ==> ['t', 'e', 'a'] ==> sorting ==> ['a', 'e', 't'] ==> string ==> "aet"
"tan" ==> ['t', 'a', 'n'] ==> sorting ==> ['a', 'n', 't'] ==> string ==> "ant"
"ate" ==> ['a', 't', 'e'] ==> sorting ==> ['a', 'e', 't'] ==> string ==> "aet"
"nat" ==> ['n', 'a', 't'] ==> sorting ==> ['a', 'n', 't'] ==> string ==> "ant"
"bat" ==> ['b', 'a', 't'] ==> sorting ==> ['a', 'b', 't'] ==> string ==> "abt"

new HashMap()

```
{
  "aet": [ "eat", "tea", "ate" ],
  "ant": [ "tan", "nat" ],
  "abt": [ "bat" ]
}
```

["eat", "tea", "ate"],
["tan", "nat"]
["bat"]

int[] marks = { 10, 30, 40, 50, 60 }

int[] marks = { }

int[] marks = new int[]

Arrays can be used to store the data, if we exactly know how many values we need to store.

int[] marks = new int[10]

abcabcbb

3

pwwkew

HashSet

abcabcbb

abcbbg