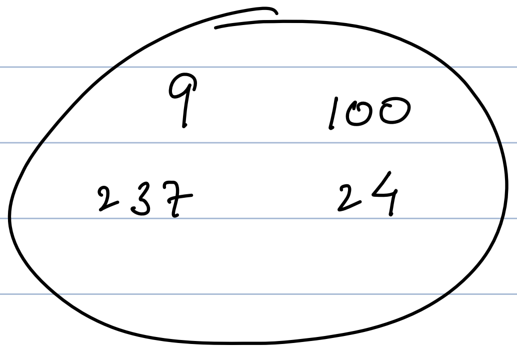
Hashset

Collection of unique elements



A = [3, 1, 2, 3, 5]

NO indexing
NO repitition

```
Hashset <Integer> hs = new Hashset<Integer>();
```

Functions →

```
hs.add(20)
hs.size()
hs.remove(331)
```

```
hs.contains(35) → true if 35 present
                  false if 35 ✗
```

All functions of HS & HM are O(1)

Q Put all elems of array into hashset

A = [1, 4, 3, -2, 1, 1, 4, 5, 3]

```
Hashset <Integer> convertToHashset (int [] arr) {

    Hashset <Integer> hs = new Hashset <Integer>();
    for ( i = 0 ; i < arr.length ; i++ ) {

            hs.add (arr [i])
    }
    return hs
}
```

Q Given 2 Hashset, print common elements

HS1 : { 0, -2, 4, 10 }   -2, 4

HS2 : { 1, -2, 5, 4, 3 }

```
void intersect( hs1 , hs2) {
    for( Integer x : hs1) {
        if( hs2. contains (x)) {
            print (x + "  ")
        }
    }
}
```

Hash map

Key – value pairs

Eg - States & their populations

"Punjab" : 5 cr
"Haryana" : 4 cr
"Delhi" : 3 cr
"UP" : 10 cr
"Maharashtra" : 10 Cr

Duplicate key NOT allowed
Same value is allowed

eg: Delhi : 10
UP : 10

Functions

```
hm.put("Delhi", 10)
hm.get("Delhi")  → 10
hm.containsKey("Punjab")
hm.remove("Delhi")
hm.size()
hm.keySet()
hm.values()  → gives all values
```

Q Given an array, return frequency hashmap

A = [1, 4, 3, -2, 1, 1, 4, 5, 3]

ans ⇒ 1 : 3
3 : 2
4 : 2
-2 : 1
5 : 1

[1, 4, 3, -2, 1, 1, 4, 5, 3]

{ 1 : ~~1~~ ~~2~~ 3
4 : ~~1~~ 2
3 : ~~1~~ 2
-2 : 1
5 : 1
}

{ 1 : ~~1~~ ~~2~~ 3
4 : ~~1~~ 2
3 : ~~1~~ 2
-2 : 1
5 : 1
}

```
Hashmap <Integer, Integer> hm= new
                         Hashmap<Integer, Integer>(),

for( i=0 ; i< arr.length ; i++ ){

     if ( hm.containskey (arr[i]) == false ){

               hm.put ( arr[i], 1)
    }

    else {
        int beforeValue = hm.get (arr[i])
        int afterValue = beforeValue +1
        hm.put (arr[i], afterValue)
    }
}
```

hm.get (key) will give you value

```
for( Integer x : hm.keySet () ){

      - — — - -
}
```

12 → 7 cr 37 lakh
14 → 1 cr - - - -

Math Array algorithms sorting & searching

Strings Interview problems

2 sum → 2 pointer
fast / slow → Linked list