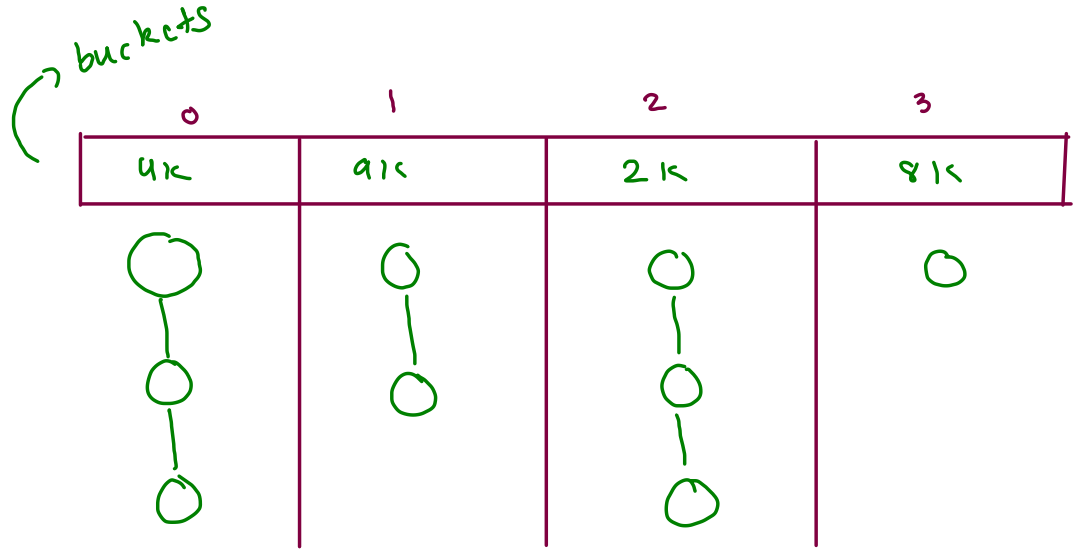


size = total nodes

```
public static class HashMap<K, V> {  
    private class HMNode {  
        K key;  
        V value;  
  
        HMNode(K key, V value) {  
            this.key = key;  
            this.value = value;  
        }  
    }  
  
    private int size; // n  
    private LinkedList<HMNode>[] buckets; // N = buckets.length  
  
    public HashMap() {  
        initbuckets(4);  
        size = 0;  
    }  
  
    private void initbuckets(int N) {  
        buckets = new LinkedList[N];  
        for (int bi = 0; bi < buckets.length; bi++) {  
            buckets[bi] = new LinkedList<>();  
        }  
    }  
}
```



0	1	2	3
<p>Ind 300</p> <p>Eng 150</p>	<p>Afg 500</p>	<p>Aus 90</p> <p>SL 200</p> <p>Africa 280</p>	<p>Nig 250</p>

map.put( Africa, 280) → g

map.put (SL, 200) → u



```

public void put(K key, V value) throws Exception {
    int bi = findBucketIndex(key);
    int di = findWithinBucket(key, bi);

    if(di == -1) {
        //key is not present in hashmap (put - insertion)
        HMNode node = new HMNode(key, value);
        buckets[bi].add(node);
        size++;
    }
    else {
        //key is present in hashmap (put - updation)
        HMNode node = buckets[bi].get(di);
        node.value = value;
    }
}

private int findBucketIndex(K key) {
    int hc = key.hashCode();

    int bi = (Math.abs(hc)) % buckets.length;

    return bi;
}

private int findWithinBucket(K key, int bi) {
    //linear search
    for(int i=0; i < buckets[bi].size(); i++) {
        HMNode node = buckets[bi].get(i);

        if(node.key.equals(key) == true) {
            return i;
        }
    }
    return -1;
}

```

0	1	2	3
<div>Ind 300</div> <div>Eng 38</div>	<div>AFg 500</div>	<div>AWs 90</div> <div>SL 200</div> <div>Africa 190</div>	<div>Nig 250</div>

Eng  $\leftrightarrow$  hashCode  $\rightarrow$  -880714  $\rightarrow$  0

map.put (Africa, 190)

map.put (Eng, 380)

0	1	2	3
○	○	○	○
○	○		○
○			○

rehashing  
is  
required

$N \rightarrow$  buckets-length

put, get, clc, remove

$$d \leq 2$$

$d =$  average no. of  
node per buckets

$$d = \frac{\text{size}}{N}$$

$$d = \frac{\text{total nodes}}{\text{total buckets}}$$

odd - buckets

0	1	2	3
Ind 300	Afg 500	Aus 90	Nig 250
Eng 38	Pak 480	SL 200	
	Eat 950	Ajira 190	

$$\text{Ind} \xrightarrow{hc} 240 \cdot 1.4 \rightarrow 0$$

$$\text{Afg} \xrightarrow{hc} 85 \cdot 1.4 \rightarrow 1$$

$$\text{SL} \xrightarrow{hc} 46 \cdot 1.4 \rightarrow 2$$

buckets

0	1	2	3	4	5	6	7
Ind 300					Afg 500	SL 200	

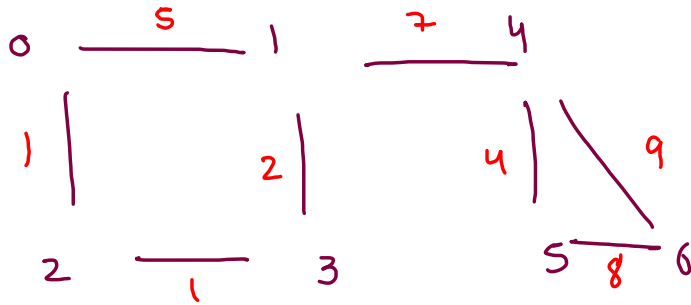
$$\text{Ind} \xrightarrow{hc} 240 \cdot 1.8 \rightarrow 0$$

$$\text{Afg} \xrightarrow{hc} 85 \cdot 1.8 \rightarrow 5$$

$$\text{SL} \xrightarrow{hc} 46 \cdot 1.8 \rightarrow 6$$

Graph:

$v, e$



[ DFS  $\rightarrow$  recursion  
BFS  $\rightarrow$  level order  
traverse

graphs

(i) shortest path — edges : BFS  
└ cost : dijkstra

(ii) conn comp

(io) ts

(iii) MST (prim)

