Gebze Technical University
Computer Engineering

CSE 222/505 – Spring 2021

Homework 5 Report

Baran Solmaz
1801042601

1. PROBLEM SOLUTION APPROACH

Part 1:

I created MapIterator class that implements Iterator class and added prev() method that return previous key.

Part 2:

I implemented HashTable with different chaining techniques(Linked List,Tree Set,Quadratic Probing)

4. Test Cases

Part 1:

```
System.out.println("--------- Part 1 ---------");
MyHashMap<Integer,String> hashMap= new MyHashMap<Integer,String>();

for (int i = 0; i < 10; i++) {
    System.out.println(".put("+i+",\""+i+"\") "+hashMap.put(i, ""+i+"" ));
}
System.out.println(hashMap);
System.out.println(".iterator(5)");
MyHashMap<Integer,String>.MapIterator iter= hashMap.iterator(5);

while (iter.hasNext()) {
    System.out.print(".next() "+iter.next()+"\t");
    System.out.println(".prev() "+iter.prev()+"\n");
}
```

Part 2:

Small =15 Medium= 1000 Large= 10000

For every HashTable class , I added small/medium/large elements. Then, I removed first half of the elements. After the deletion process, I changed second half of elements with different values. In the end, I tested get method with exist and non-exist elements.

5. Running Commands and Results

Part 1:

```
--------- Part 1 ---------
.put(0,"0") null
.put(1,"1") null
.put(2,"2") null
.put(3,"3") null
.put(4,"4") null
.put(5,"5") null
.put(6,"6") null
.put(7,"7") null
.put(8,"8") null
.put(9,"9") null
{0=0, 1=1, 2=2, 3=3, 4=4, 5=5, 6=6, 7=7, 8=8, 9=9}
.iterator(5)
.next() 5       .prev() 5

.next() 6       .prev() 6

.next() 7       .prev() 7

.next() 8       .prev() 8

.next() 9       .prev() 9

.next() 0       .prev() 0

.next() 1       .prev() 1

.next() 2       .prev() 2

.next() 3       .prev() 3

.next() 4       .prev() 4
```

Part 2:

```
--------- Part 2 ---------

1) Chaing by Using Linked List

1-a) Small Data Size :15
.put(0,"0") null
.put(2,"1") null
.put(4,"2") null
.put(6,"3") null
.put(8,"4") null
.put(10,"5") null
.put(12,"6") null
.put(14,"7") null
.put(16,"8") null
.put(18,"9") null
.put(20,"10") null
.put(22,"11") null
.put(24,"12") null
.put(26,"13") null
.put(28,"14") null
Time : 2.244185 ms.
```

```
[[0=0] [2=1] [4=2] [6=3] [8=4] [10=5] [12=6] [14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13
] [28=14] ]
.remove(0) 0
.remove(2) 1
.remove(4) 2
.remove(6) 3
.remove(8) 4
.remove(10) 5
.remove(12) 6
Time: 7.215515 ms.
```

```
[[14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13] [28=14] ]
.put(14,"21") 7
.put(16,"24") 8
.put(18,"27") 9
.put(20,"30") 10
.put(22,"33") 11
.put(24,"36") 12
.put(26,"39") 13
.put(28,"42") 14
Time: 1.247526 ms.
```

```
[[14=21] [16=24] [18=27] [20=30] [22=33] [24=36] [26=39] [28=42] ]
.get(16) 24
.get(18) 27
.get(20) 30
.get(22) 33
.get(24) 36
.get(26) 39
.get(28) 42
Time: 0.875914 ms.
```

```
[[14=21] [16=24] [18=27] [20=30] [22=33] [24=36] [26=39] [28=42] ]
.get(-10) null
.get(-9) null
.get(-8) null
.get(-7) null
.get(-6) null
.get(-5) null
.get(-4) null
.get(-3) null
.get(-2) null
.get(-1) null
Time: 0.930594 ms.
```

```
1-b) Medium Data Size :1000
0 to 999 added
Time: 6.4819 ms.
0 to 499 removed
Time: 0.650698 ms.
500 to 999 every even number added with different values
Time: 0.493105 ms.
501 to 999 every odd number gotten with get() method
Time: 0.127363 ms.
-10 to 0 every number gotten with get() method --Non-Exist
Time: 0.006774 ms.
```

```
1-c) Large Data Size :10000
0 to 9999 added
Time: 13.071696 ms.
0 to 4999 removed
Time: 2.583305 ms.
5000 to 9999 every even number added with different values
Time: 2.103847 ms.
5001 to 9999 every odd number gotten with get() method
Time: 1.071827 ms.
-10 to 0 every number gotten with get() method --Non-Exist
Time: 0.014005 ms.
```

```
2) Chaing by Using Tree Set

2-a) Small Data Size :15

.put(0,"0") null
.put(2,"1") null
.put(4,"2") null
.put(6,"3") null
.put(8,"4") null
.put(10,"5") null
.put(12,"6") null
.put(14,"7") null
.put(16,"8") null
.put(18,"9") null
.put(20,"10") null
.put(22,"11") null
.put(24,"12") null
.put(26,"13") null
.put(28,"14") null
Time: 2.863529 ms.
```

```
[[0=0] [2=1] [4=2] [6=3] [8=4] [10=5] [12=6] [14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13
] [28=14] ]
.remove(0) 0
.remove(2) 1
.remove(4) 2
.remove(6) 3
.remove(8) 4
.remove(10) 5
.remove(12) 6
Time: 0.679979 ms.
```

```
[[14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13] [28=14] ]
.put(14,"7") 7
.put(16,"8") 8
.put(18,"9") 9
.put(20,"10") 10
.put(22,"11") 11
.put(24,"12") 12
.put(26,"13") 13
.put(28,"14") 14
Time: 1.039277 ms.
```

```
[[14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13] [28=14] ]
.get(16) 8
.get(18) 9
.get(20) 10
.get(22) 11
.get(24) 12
.get(26) 13
.get(28) 14
Time: 0.656187 ms.
```

```
[[14=7] [16=8] [18=9] [20=10] [22=11] [24=12] [26=13] [28=14] ]
.get(-10) null
.get(-9) null
.get(-8) null
.get(-7) null
.get(-6) null
.get(-5) null
.get(-4) null
.get(-3) null
.get(-2) null
.get(-1) null
Time: 0.739732 ms.
```

```
2-b) Medium Data Size :1000
0 to 999 added
Time: 6.815692 ms.
0 to 499 removed
Time: 0.861641 ms.
500 to 999 every even number added with different values
Time: 0.354451 ms.
501 to 999 every odd number gotten with get() method
Time: 0.137639 ms.
-10 to 0 every number gotten with get() method ---Non-Exist
Time: 0.00778 ms.
```

```
2-c) Large Data Size :10000
0 to 9999 added
Time: 10.568431 ms.
0 to 4999 removed
Time: 3.009811 ms.
5000 to 9999 every even number added with different values
Time: 1.605773 ms.
5001 to 9999 every odd number gotten with get() method
Time: 1.195225 ms.
-10 to 0 every number gotten with get() method ---Non-Exist
Time: 0.005016 ms.
```

```
3) Chaing by Using Tree Set

3-a) Small Data Size :15

.put(0,"0") null
.put(2,"1") null
.put(4,"2") null
.put(6,"3") null
.put(8,"4") null
.put(10,"5") null
.put(12,"6") null
.put(14,"7") null
.put(16,"8") null
.put(18,"9") null
.put(20,"10") null
.put(22,"11") null
.put(24,"12") null
.put(26,"13") null
.put(28,"14") null
Time: 1.329483 ms.
```

Hash Value	Key	Next
0	0=0	1
1	20=10	Null
2	2=1	3
3	22=11	Null
4	4=2	5
5	24=12	Null
6	6=3	7
7	26=13	Null
8	8=4	9
9	28=14	Null
10	10=5	Null
11		Null
12	12=6	Null
13		Null
14	14=7	Null
15		Null
16	16=8	Null
17		Null
18	18=9	Null
19		Null

```
.remove(0) 0
.remove(2) 1
.remove(4) 2
.remove(6) 3
.remove(8) 4
.remove(10) 5
.remove(12) 6
Time: 0.531415 ms.
```

Hash Value	Key	Next
0	20=10	Null
1		Null
2	22=11	Null
3		Null
4	24=12	Null
5		Null
6	26=13	Null
7		Null
8	28=14	Null
9		Null
10		Null
11		Null
12		Null
13		Null
14	14=7	Null
15		Null
16	16=8	Null
17		Null
18	18=9	Null
19		Null

```
.put(14,"21") 7
.put(16,"24") 8
.put(18,"27") 9
.put(20,"30") 10
.put(22,"33") 11
.put(24,"36") 12
.put(26,"39") 13
.put(28,"42") 14
Time: 0.822348 ms.
```

```
Hash
Value   Key     Next

 0      20=30   Null
 1              Null
 2      22=33   Null
 3              Null
 4      24=36   Null
 5              Null
 6      26=39   Null
 7              Null
 8      28=42   Null
 9              Null
 10             Null
 11             Null
 12             Null
 13             Null
 14     14=21   Null
 15             Null
 16     16=24   Null
 17             Null
 18     18=27   Null
 19             Null
```

```
.get(16) 24
.get(18) 27
.get(20) 30
.get(22) 33
.get(24) 36
.get(26) 39
.get(28) 42
Time: 0.53065 ms.
```

```
Hash
Value   Key     Next

 0      20=30   Null
 1              Null
 2      22=33   Null
 3              Null
 4      24=36   Null
 5              Null
 6      26=39   Null
 7              Null
 8      28=42   Null
 9              Null
 10             Null
 11             Null
 12             Null
 13             Null
 14     14=21   Null
 15             Null
 16     16=24   Null
 17             Null
 18     18=27   Null
 19             Null
```

```
.get(-10) null
.get(-9) null
.get(-8) null
.get(-7) null
.get(-6) null
.get(-5) null
.get(-4) null
.get(-3) null
.get(-2) null
.get(-1) null
Time: 0.599604 ms.
```

```
Hash
Value   Key     Next

 0      20=30   Null
 1              Null
 2      22=33   Null
 3              Null
 4      24=36   Null
 5              Null
 6      26=39   Null
 7              Null
 8      28=42   Null
 9              Null
 10             Null
 11             Null
 12             Null
 13             Null
 14     14=21   Null
 15             Null
 16     16=24   Null
 17             Null
 18     18=27   Null
 19             Null
```

```
3-b) Medium Data Size :1000
0 to 999 added
Time: 1.297376 ms.
0 to 499 removed
Time: 0.131207 ms.
500 to 999 every even number added with different values
Time: 2.660023 ms.
501 to 999 every odd number gotten with get() method
Time: 0.054908 ms.
-10 to 0 every number gotten with get() method ---Non-Exist
Time: 0.003554 ms.
```

```
3-c) Large Data Size :10000
0 to 9999 added
Time: 13.323757 ms.
0 to 4999 removed
Time: 1.539068 ms.
5000 to 9999 every even number added with different values
Time: 65.81885 ms.
5001 to 9999 every odd number gotten with get() method
Time: 0.582804 ms.
-10 to 0 every odd number gotten with get() method --Non-Exist
Time: 1.416167 ms.
```