

GIT Department of Computer Engineering

**CSE 222/505 - Spring 2021
Homework 5 # Report**

Okan Torun

1801042662

PART1

1)PROBLEM SOLUTION APPROACH

Here, we were asked to apply commands such as next, prev, hasNext on the hash map with the iterator. Since Hash map does not have a get method, I moved on the list with foreach and returned the desired elements. And with the help of the counters I used, I used a circular iterator. I used traversalCount,currentSize and size variables. traversalCount enabled me to traverse on the map in a circular fashion. I made it possible to start the iterator object that I created in the driver class with the HashMapIter class.

If I complete the circular traverse, the next element I will rotate is the element before the starting point. If I get to the point where I started by making prev, I designed it to return the starting point with prev and next.

2)TEST CASES

-I'm adding a hashmap element.

```
//ADDING ELEMENT
HashMap<Integer, String> obj=new HashMap<>();
obj.put(1, "Ocak");
obj.put(2, "Şubat");
obj.put(3, "Mart");
obj.put(4, "Nisan");
obj.put(5, "Mayıs");
obj.put(6, "Haziran");
obj.put(7, "Temmuz");
obj.put(8, "Ağustos");
```

I am starting from the iterator 3rd element.

```
//START FROM GIVEN KEY
MapIterator<Integer, String> iter=obj.iterator(3);
```

The iterator is started from the beginning.

```
//OR START FROM BEGINNING
MapIterator<Integer, String> iter=obj.iterator();
```

With the iterator, I proceed to the end of the list with the next method.

```
System.out.println("-----TEST FOR NEXT METHOD-----");
while(iter.hasNext())
    System.out.println("Next-->" + iter.next());
```

I check if the next element can be reached.

```
System.out.println("\n-----TEST FOR hasNext METHOD-----");
System.out.println("hasNext-->" + iter.hasNext());
```

I'm going to the starting point with the prev method.

```

System.out.println("\n-----TEST FOR PREV METHOD-----");
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());
System.out.println("Prev-->" + iter.prev());

```

3)RUNNING AND RESULTS

```

|-----TEST FOR NEXT METHOD-----
Next-->3
Next-->4
Next-->5
Next-->6
Next-->7
Next-->8
Next-->1
Next-->2
Next-->2

-----TEST FOR hasNext METHOD-----
hasNext-->>false

-----TEST FOR PREV METHOD-----
Prev-->2
Prev-->1
Prev-->8
Prev-->7
Prev-->6
Prev-->5
Prev-->4
Prev-->3
Prev-->3

-----TEST FOR hasNext METHOD-----
hasNext-->>true

```

PART2

1)PROBLEM SOLUTION APPROACH

Here we have discussed 3 different uses of the hashmap data structure. These are tableTreeSet, tableLinkedList and tableCoalesced. We measured the efficiency on these data structures by adding, deleting and connecting methods. The most efficient of these is tableCoalesced. We used the method of keeping colliding elements of the public addressing structure. At the same time, the indices that the elements will go to were determined in a quadratic way. I saved the end position index of the element to the next variable of the last colliding element before the index it went to. More important than losing the index of colliding elements here is that this chaining continues while the element is deleted. If there is an element, I performed the scrolling in the loop. And I continued this process until the chain was over. Next, I tried to avoid any possible confusion by making the next element in the last link of the chain empty. I have successfully performed the deletion and addition operations with these operations.

2)TEST CASES

I add 10 elements to each structure for the little table in the beginnig.

```
for(int i=0;i<10;i++) {  
    tableBook.put(i, i);  
    tableCoalesced.put(i, i);  
    tableTreeSet.put(i, i);  
}
```

I add an element to each element in each loop and calculate the elapsed time.

```
System.out.println("-----SMALL TABLE-----");  
System.out.println("ADD-----");  
for(int i=0;i<5;i++) {  
    startTime = System.nanoTime();  
    tableTreeSet.put(11+i, i);  
    endTime = System.nanoTime();  
    time = endTime - startTime;  
    System.out.println("Time Spent For the TreeSet when adding one element->>" + time);  
    startTime = System.nanoTime();  
    tableBook.put(11+i, i);  
    endTime = System.nanoTime();  
    time = endTime - startTime;  
    System.out.println("Time Spent For the LinkedList when adding one element->>" + time);  
    startTime = System.nanoTime();  
    tableCoalesced.put(11+i, i);  
    endTime = System.nanoTime();  
    time = endTime - startTime;  
    System.out.println("Time Spent For the Coalesced when adding one element->>" + time);  
    System.out.println("----");  
}
```

I try to access an element respectively in each loop and calculate the elapsed time.

```
System.out.println("GET exist-----");
for(int i=0;i<5;i++) {
    startTime = System.nanoTime();
    tableTreeSet.get(i+1);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the TreeSet when accessing element->>" + time);
    startTime = System.nanoTime();
    tableBook.get(i+1);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the LinkedList when accessing element->>" + time);
    startTime = System.nanoTime();
    tableCoalesced.get(i+1);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the Coalesced when accessing element->>" + time);
    System.out.println("----");
}
```

I am trying to access the non-existing element in each loop and calculate the elapsed time.

```
System.out.println("GET non-exist-----");
for(int i=0;i<5;i++) {
    startTime = System.nanoTime();
    tableTreeSet.get(100+i);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the TreeSet when accessing non-exist element->>" + time);
    startTime = System.nanoTime();
    tableBook.get(100+i);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the LinkedList when accessing non-exist element->>" + time);
    startTime = System.nanoTime();
    tableCoalesced.get(100+i);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the Coaledced when accessing non-exist element->>" + time);
    System.out.println("----");
}
```


I am trying to delete random elements in each loop and calculate the elapsed time.

```
System.out.println("REMOVE-----");
for(int i=0;i<5;i++) {
    startTime = System.nanoTime();
    tableTreeSet.remove(i+5);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the TreeSet when removing one element->>" + time);
    startTime = System.nanoTime();
    tableBook.remove(i+5);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the LinkedList when removing one element->>" + time);
    startTime = System.nanoTime();
    tableCoalesced.remove(i+5);
    endTime = System.nanoTime();
    time = endTime - startTime;
    System.out.println("Time Spent For the Coalesced when removing one element->>" + time);
    System.out.println("----");
}
```

3)RUNNING AND RESULTS

```
-----SMALL TABLE-----
ADD-----
Time Spent For the TreeSet when adding one element->>18600
Time Spent For the LinkedList when adding one element->>12800
Time Spent For the Coalesced when adding one element->>2100
----
Time Spent For the TreeSet when adding one element->>5800
Time Spent For the LinkedList when adding one element->>3300
Time Spent For the Coalesced when adding one element->>1300
----
Time Spent For the TreeSet when adding one element->>4300
Time Spent For the LinkedList when adding one element->>2800
Time Spent For the Coalesced when adding one element->>1600
----
Time Spent For the TreeSet when adding one element->>11900
Time Spent For the LinkedList when adding one element->>3200
Time Spent For the Coalesced when adding one element->>1300
----
Time Spent For the TreeSet when adding one element->>4300
Time Spent For the LinkedList when adding one element->>2800
Time Spent For the Coalesced when adding one element->>1100
----
GET exist-----
Time Spent For the TreeSet when accessing element->>5800
Time Spent For the LinkedList when accessing element->>4100
Time Spent For the Coalesced when accessing element->>3100
----
Time Spent For the TreeSet when accessing element->>2200
Time Spent For the LinkedList when accessing element->>2400
Time Spent For the Coalesced when accessing element->>1000
----
Time Spent For the TreeSet when accessing element->>2300
Time Spent For the LinkedList when accessing element->>2600
Time Spent For the Coalesced when accessing element->>900
----
Time Spent For the TreeSet when accessing element->>1900
Time Spent For the LinkedList when accessing element->>2400
Time Spent For the Coalesced when accessing element->>900
----
Time Spent For the TreeSet when accessing element->>2000
Time Spent For the LinkedList when accessing element->>2300
Time Spent For the Coalesced when accessing element->>1000
----
```



```

GET non-exist-----
Time Spent For the TreeSet when accessing non-exist element-->2600
Time Spent For the LinkedList when accessing non-exist element-->2300
Time Spent For the Coalecdced when accessing non-exist element-->800
----
Time Spent For the TreeSet when accessing non-exist element-->2800
Time Spent For the LinkedList when accessing non-exist element-->2600
Time Spent For the Coalecdced when accessing non-exist element-->900
----
Time Spent For the TreeSet when accessing non-exist element-->2700
Time Spent For the LinkedList when accessing non-exist element-->2500
Time Spent For the Coalecdced when accessing non-exist element-->700
----
Time Spent For the TreeSet when accessing non-exist element-->2400
Time Spent For the LinkedList when accessing non-exist element-->2700
Time Spent For the Coalecdced when accessing non-exist element-->700
----
Time Spent For the TreeSet when accessing non-exist element-->2600
Time Spent For the LinkedList when accessing non-exist element-->3200
Time Spent For the Coalecdced when accessing non-exist element-->700
----

```

```

REMOVE-----
Time Spent For the TreeSet when removing one element-->16900
Time Spent For the LinkedList when removing one element-->11000
Time Spent For the Coalesced when removing one element-->3700
----
Time Spent For the TreeSet when removing one element-->4200
Time Spent For the LinkedList when removing one element-->3200
Time Spent For the Coalesced when removing one element-->1400
----
Time Spent For the TreeSet when removing one element-->3700
Time Spent For the LinkedList when removing one element-->2900
Time Spent For the Coalesced when removing one element-->1400
----
Time Spent For the TreeSet when removing one element-->3600
Time Spent For the LinkedList when removing one element-->2900
Time Spent For the Coalesced when removing one element-->1400
----
Time Spent For the TreeSet when removing one element-->3600
Time Spent For the LinkedList when removing one element-->2900
Time Spent For the Coalesced when removing one element-->1300
----

```

```

-----MEDIUM TABLE-----
ADD-----
Time Spent For the TreeSet when adding one element-->5200
Time Spent For the LinkedList when adding one element-->6500
Time Spent For the Coalesced when adding one element-->4600
----
Time Spent For the TreeSet when adding one element-->5800
Time Spent For the LinkedList when adding one element-->3900
Time Spent For the Coalesced when adding one element-->1200
----
Time Spent For the TreeSet when adding one element-->9300
Time Spent For the LinkedList when adding one element-->2000
Time Spent For the Coalesced when adding one element-->1000
----
Time Spent For the TreeSet when adding one element-->5900
Time Spent For the LinkedList when adding one element-->3500
Time Spent For the Coalesced when adding one element-->1000
----
Time Spent For the TreeSet when adding one element-->6800
Time Spent For the LinkedList when adding one element-->6500
Time Spent For the Coalesced when adding one element-->1000
----

```

```

GET exist-----
Time Spent For the TreeSet when accessing element-->>1600
Time Spent For the LinkedList when accessing element-->>3000
Time Spent For the Coalesced when accessing element-->>6900
----
Time Spent For the TreeSet when accessing element-->>1200
Time Spent For the LinkedList when accessing element-->>2300
Time Spent For the Coalesced when accessing element-->>900
----
Time Spent For the TreeSet when accessing element-->>2200
Time Spent For the LinkedList when accessing element-->>3200
Time Spent For the Coalesced when accessing element-->>800
----
Time Spent For the TreeSet when accessing element-->>2100
Time Spent For the LinkedList when accessing element-->>3900
Time Spent For the Coalesced when accessing element-->>700
----
Time Spent For the TreeSet when accessing element-->>2100
Time Spent For the LinkedList when accessing element-->>3300
Time Spent For the Coalesced when accessing element-->>800

GET non-exist-----
Time Spent For the TreeSet when removing one element-->>2300
Time Spent For the LinkedList when accessing non-exist element-->>2900
Time Spent For the Coaledced when accessing non-exist element-->>3100
----
Time Spent For the TreeSet when removing one element-->>2500
Time Spent For the LinkedList when accessing non-exist element-->>2700
Time Spent For the Coaledced when accessing non-exist element-->>2900
----
Time Spent For the TreeSet when removing one element-->>2000
Time Spent For the LinkedList when accessing non-exist element-->>3000
Time Spent For the Coaledced when accessing non-exist element-->>2700
----
Time Spent For the TreeSet when removing one element-->>2600
Time Spent For the LinkedList when accessing non-exist element-->>2600
Time Spent For the Coaledced when accessing non-exist element-->>2600
----
Time Spent For the TreeSet when removing one element-->>2200
Time Spent For the LinkedList when accessing non-exist element-->>2800
Time Spent For the Coaledced when accessing non-exist element-->>3100
----

REMOVE-----
Time Spent For the TreeSet when removing one element-->>5900
Time Spent For the LinkedList when removing one element-->>5900
Time Spent For the Coalesced when removing one element-->>1400
----
Time Spent For the TreeSet when removing one element-->>4100
Time Spent For the LinkedList when removing one element-->>5000
Time Spent For the Coalesced when removing one element-->>1200
----
Time Spent For the TreeSet when removing one element-->>4000
Time Spent For the LinkedList when removing one element-->>6200
Time Spent For the Coalesced when removing one element-->>1100
----
Time Spent For the TreeSet when removing one element-->>3600
Time Spent For the LinkedList when removing one element-->>4300
Time Spent For the Coalesced when removing one element-->>1300
----
Time Spent For the TreeSet when removing one element-->>4600
Time Spent For the LinkedList when removing one element-->>5700
Time Spent For the Coalesced when removing one element-->>1800
----

```

-----LARGE TABLE-----

ADD-----

Time Spent For the TreeSet when adding one element->>1400
Time Spent For the LinkedList when adding one element->>3600
Time Spent For the Coalesced when adding one element->>700

Time Spent For the TreeSet when adding one element->>1000
Time Spent For the LinkedList when adding one element->>700
Time Spent For the Coalesced when adding one element->>400

Time Spent For the TreeSet when adding one element->>1224600
Time Spent For the LinkedList when adding one element->>1600
Time Spent For the Coalesced when adding one element->>900

Time Spent For the TreeSet when adding one element->>2700
Time Spent For the LinkedList when adding one element->>1000
Time Spent For the Coalesced when adding one element->>500

Time Spent For the TreeSet when adding one element->>6200
Time Spent For the LinkedList when adding one element->>1000
Time Spent For the Coalesced when adding one element->>1300

GET exist-----

Time Spent For the TreeSet when accessing element->>4400
Time Spent For the LinkedList when accessing element->>22600
Time Spent For the Coalesced when accessing element->>12600

Time Spent For the TreeSet when accessing element->>2600
Time Spent For the LinkedList when accessing element->>2100
Time Spent For the Coalesced when accessing element->>10200

Time Spent For the TreeSet when accessing element->>2100
Time Spent For the LinkedList when accessing element->>1500
Time Spent For the Coalesced when accessing element->>1000

Time Spent For the TreeSet when accessing element->>6000
Time Spent For the LinkedList when accessing element->>1800
Time Spent For the Coalesced when accessing element->>900

Time Spent For the TreeSet when accessing element->>1800
Time Spent For the LinkedList when accessing element->>1900
Time Spent For the Coalesced when accessing element->>800

GET non-exist-----

Time Spent For the TreeSet when removing one element->>3000
Time Spent For the LinkedList when accessing non-exist element->>2400
Time Spent For the Coaledced when accessing non-exist element->>1000

Time Spent For the TreeSet when removing one element->>2200
Time Spent For the LinkedList when accessing non-exist element->>2100
Time Spent For the Coaledced when accessing non-exist element->>700

Time Spent For the TreeSet when removing one element->>2200
Time Spent For the LinkedList when accessing non-exist element->>2100
Time Spent For the Coaledced when accessing non-exist element->>700

Time Spent For the TreeSet when removing one element->>2400
Time Spent For the LinkedList when accessing non-exist element->>2100
Time Spent For the Coaledced when accessing non-exist element->>900

Time Spent For the TreeSet when removing one element->>2200
Time Spent For the LinkedList when accessing non-exist element->>1500
Time Spent For the Coaledced when accessing non-exist element->>700

```
REMOVE-----
Time Spent For the TreeSet when removing one element->>8400
Time Spent For the LinkedList when removing one element->>4300
Time Spent For the Coalesced when removing one element->>2100
-----
Time Spent For the TreeSet when removing one element->>8600
Time Spent For the LinkedList when removing one element->>3900
Time Spent For the Coalesced when removing one element->>1500
-----
Time Spent For the TreeSet when removing one element->>4100
Time Spent For the LinkedList when removing one element->>3400
Time Spent For the Coalesced when removing one element->>1500
-----
Time Spent For the TreeSet when removing one element->>4600
Time Spent For the LinkedList when removing one element->>3300
Time Spent For the Coalesced when removing one element->>1400
-----
Time Spent For the TreeSet when removing one element->>4400
Time Spent For the LinkedList when removing one element->>3300
Time Spent For the Coalesced when removing one element->>1300
-----
```