

**GIT Department of Computer Engineering
CSE 222/505 - Spring 2021
Homework 5 Report**

**Student Name
Samet NALBANT
Student Number
1801042614**

1. PROBLEM SOLUTION APPROACH

For the part1, I create a MyHashMap class which extends from Java util's HashMap. So this way I could use necessary functions of HashMap. To design iterator I create a Iterator interface. I implement this interface to my Iterator Class which is nested class of MyHashMap class. After that point I create necessary data fields such as index, next value etc. Than I implement the functions which are defined on homework file.

For the part2, I create three class: HashTableChain, HashTable and CoalescedHashMap. I implement the KWHHashMap interface to all of them. I take a reference from book for the HashTable class which implements linked list. After that I create HashTableChain using TreeSet. Finally I create the CoalescedHashMap class and I implement the necessary function in it.

2. TEST CASES

Part1)

I create HashMap which hold programming languages and their developed year's.

```
MyHashMap<String, Integer> languages = new MyHashMap<String, Integer>();
languages.put("C", 1972);
languages.put("C++", 1983);
languages.put("JAVA", 1995);
languages.put("PYTHON", 1990);
languages.put("RUBY", 1995);
languages.put("KOTLIN", 2011);
```

I create a Iterator from the C++ language. Then I print the keys using Iterator's next method.

```
System.out.println("Iterator is creating from C++!");
Iterator<String> temp = languages.iterator("C++");
System.out.println("Next values of C++ are printing using Iterator's methods.");
while(temp.hasNext()) {
    System.out.println(temp.next());
}
```

I take the position to C++ again and I print the previous keys using Iterator's prev method.

```
temp = languages.iterator("C++");
System.out.println("-----");
System.out.println("Prev values are printing using Iterator's methods.");
while(temp.hasPrev()) {
    System.out.println(temp.prev());
}
```

I create Iterator from not existed value. Then I print keys using Iterator's next method.

```
System.out.println("Iterator is creating from not existed key.");
temp = languages.iterator("C#");
System.out.println("Next values of Iterator are printing using Iterator's methods.");
while(temp.hasNext()) {
    System.out.println(temp.next());
}
```

Part2)

I create a HashTableChain, HashTableChainTreeSet, CoalescedHashMap. Then I put them 10 different values.

```
HashTableChain<Integer, Integer> data1 = new HashTableChain<Integer, Integer>();
HashTableChainTreeSet<Integer, Integer> data2 = new HashTableChainTreeSet<Integer, Integer>();
CoalescedHashMap<Integer, Integer> data3 = new CoalescedHashMap<Integer, Integer>();
long start;
long end;
System.out.println("-----Small Data-----");
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.put(i,i);
}
end =System.nanoTime();
System.out.println("Put time for HashTableChain with different values: "+ (end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.put(i,i);
}
end =System.nanoTime();
System.out.println("Put time for HashTableChainTreeSet with different values: "+ (end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data3.put(i,i);
}
end =System.nanoTime();
System.out.println("Put time for CoalescedHashMap with different values: "+ (end-start));
```

I put them 10 same values.

```
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.put(2,2);
}
end =System.nanoTime();
System.out.println("Put time for HashTableChain with same values: "+ (end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.put(2,2);
}
end =System.nanoTime();
System.out.println("Put time for HashTableChainTreeSet with same values: "+ (end-start));
for(int i=0;i<10;i++) {
    data3.put(2,2);
}
end =System.nanoTime();
System.out.println("Put time for CoalescedHashMap with same values: "+ (end-start));
```

I call get method with exist values 10 times.

```
System.out.println("-----");
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.get(i);
}
end = System.nanoTime();
System.out.println("Get time for HashTableChain with different values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.get(i);
}
end = System.nanoTime();
System.out.println("Get time for HashTableChainTreeSet with different values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data3.get(i);
}
end = System.nanoTime();
System.out.println("Get time for CoalescedHashMap with different values: "+(end-start));
```

I call get method with non existing values 10 times.

```
System.out.println("-----");
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.get(99);
}
end = System.nanoTime();
System.out.println("Get time for HashTableChain with non existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.get(99);
}
end = System.nanoTime();
System.out.println("Get time for HashTableChainTreeSet with non existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data3.get(99);
}
end = System.nanoTime();
System.out.println("Get time for CoalescedHashMap with non existing values: "+(end-start));
```

I call remove method with 10 values.

```
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.remove(i);
}
end = System.nanoTime();
System.out.println("Remove time for HashTableChain with existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.remove(i);
}
end = System.nanoTime();
System.out.println("Remove time for HashTableChainTreeSet with existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data3.remove(i);
}
end = System.nanoTime();
System.out.println("Remove time for CoalescedHashMap with existing values: "+(end-start));
```

I call remove method with 10 non existing values.

```
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data1.remove(555);
}
end = System.nanoTime();
System.out.println("Remove time for HashTableChain with non existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data2.remove(555);
}
end = System.nanoTime();
System.out.println("Remove time for HashTableChainTreeSet with non existing values: "+(end-start));
start = System.nanoTime();
for(int i=0;i<10;i++) {
    data3.remove(555);
}
end = System.nanoTime();
System.out.println("Remove time for CoalescedHashMap with non existing values: "+(end-start));
```

I used the same parts for the medium and larged data. For the medium I based on 100 and for the large size I based on 1000 values.

3. RUNNING AND RESULTS

PART1)

```
-----Part 1 Test Started!-----
Hash map which contains programming languages as a key and their developing dates as a value is creating!
-----Keys-----
JAVA
C++
C
RUBY
PYTHON
KOTLIN
-----
Iterator is creating from C++!
Next values of C++ are printing using Iterator's methods.
C++
C
RUBY
PYTHON
KOTLIN
JAVA
-----
```

```
Prev values are printing using Iterator's methods.
JAVA
-----
Iterator is creating from not existed key.
Next values of Iterator are printing using Iterator's methods.
JAVA
C++
C
RUBY
PYTHON
KOTLIN
-----
```

PART2)

```
-----Part 2 Test Started!-----
-----Small Data-----
Put time for HashTableChain with different values: 716700
Put time for HashTableChainTreeSet with different values: 1032100
Put time for CoalescedHashMap with different values: 16800
-----
Put time for HashTableChain with same values: 20700
Put time for HashTableChainTreeSet with same values: 18700
Put time for CoalescedHashMap with same values: 47600
-----
Get time for HashTableChain with different values: 9800
Get time for HashTableChainTreeSet with different values: 10900
Get time for CoalescedHashMap with different values: 6800
-----
Get time for HashTableChain with non existing values: 1800
Get time for HashTableChainTreeSet with non existing values: 1600
Get time for CoalescedHashMap with non existing values: 2200
-----
Remove time for HashTableChain with existing values: 19600
Remove time for HashTableChainTreeSet with existing values: 29100
Remove time for CoalescedHashMap with existing values: 8500
-----
Remove time for HashTableChain with non existing values: 10300
Remove time for HashTableChainTreeSet with non existing values: 2700
Remove time for CoalescedHashMap with non existing values: 9700
-----
```

-----Medium Data-----

Put time for HashTableChain with different values: 195300
Put time for HashTableChainTreeSet with different values: 197000
Put time for CoalescedHashMap with different values: 107800

Put time for HashTableChain with same values: 234700
Put time for HashTableChainTreeSet with same values: 425100
Put time for CoalescedHashMap with same values: 616600

Get time for HashTableChain with different values: 250600
Get time for HashTableChainTreeSet with different values: 205800
Get time for CoalescedHashMap with different values: 50900

Get time for HashTableChain with non existing values: 111000
Get time for HashTableChainTreeSet with non existing values: 95100
Get time for CoalescedHashMap with non existing values: 30100

Remove time for HashTableChain with existing values: 118500
Remove time for HashTableChainTreeSet with existing values: 226200
Remove time for CoalescedHashMap with existing values: 20400

Remove time for HashTableChain with non existing values: 8500
Remove time for HashTableChainTreeSet with non existing values: 6600
Remove time for CoalescedHashMap with non existing values: 28400

-----Large Data-----

Put time for HashTableChain with different values: 1458800
Put time for HashTableChainTreeSet with different values: 4215300
Put time for CoalescedHashMap with different values: 206400

Put time for HashTableChain with same values: 199200
Put time for HashTableChainTreeSet with same values: 597500
Put time for CoalescedHashMap with same values: 718400

Get time for HashTableChain with different values: 582800
Get time for HashTableChainTreeSet with different values: 1107900
Get time for CoalescedHashMap with different values: 95400

Get time for HashTableChain with non existing values: 154400
Get time for HashTableChainTreeSet with non existing values: 136500
Get time for CoalescedHashMap with non existing values: 95000

Remove time for HashTableChain with existing values: 836200
Remove time for HashTableChainTreeSet with existing values: 1001500
Remove time for CoalescedHashMap with existing values: 161400

Remove time for HashTableChain with non existing values: 64400
Remove time for HashTableChainTreeSet with non existing values: 65500
Remove time for CoalescedHashMap with non existing values: 375100
