

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 7 REPORT

Deniz Can Erdem Yılmaz
151044001

Course Assistant: Şeyma Yücer

5. Problem Solutions Approach

To build a navigable map a binary search tree has used as a container for BinaryNavMap. Although it is easy to use a binary search tree for insertion, extraction and search operations I found traversing the tree through an iterator much easier for this homework. In almost every method an treeIterator object created to use to iterate each object to do comparisons.

In second part to build a chained hash table an open adressed hash table used as container instead of a single linked list. To be honest I don't truly understand the reason and probably couldn't build the structure as its desired form.

6. Test Cases

Each class' methods tested inside the main test methods that given from TA. Also some other tests made to be sure about methods. BinaryNavMap works fine except 3 un-implemented methods. Tested with elements inside and outside the map. Due to lack of inputs there are not much error cases to handle and test.

7. Running and Results

```
The original set odds is {gebze=kocaeli, uskudar=istanbul, kahta=adiyaman, niksar=tokat, foca=izmir, cekirge=bursa, kecioren=ankara, aksaray=istanbul, kadikoy=istanbul, biga=canakkale, manavgat=antalya}
The ordered set m is {uskudar=istanbul}
Set of navigable: [gebze --> kocaeli, uskudar --> istanbul, kahta --> adiyaman, niksar --> tokat, foca --> izmir, cekirge --> bursa, kecioren --> ankara, aksaray --> istanbul, kadikoy --> istanbul, biga --> canakkale, manavgat=antalya]

The first entry is aksaray --> istanbul
The first key is aksaray

The last entry is uskudar --> istanbul
The last key is uskudar

Ceiling entry of foca is: foca --> izmir
Ceiling key of foca is: foca

Higher entry of foca is: gebze --> kocaeli
Higher key of foca is: gebze

Floor entry of foca is: foca --> izmir
Floor key of foca is: foca

Lower entry of foca is: cekirge --> bursa
Lower key of foca is: cekirge

Tail map of foca is: {kecioren=ankara, kahta=adiyaman, foca=izmir, gebze=kocaeli, kadikoy=istanbul, manavgat=antalya, niksar=tokat, uskudar=istanbul}
Head map of foca is: {biga=canakkale, cekirge=bursa, aksaray=istanbul}

Poll first: aksaray --> istanbul
Poll last: uskudar --> istanbul
null
istanbul
null
17
Your tests is done. Make sure that you test all methods of class!!

Process finished with exit code 0
```

```

19
20 public static Boolean Q1Test()
21 {
22     String testing = "foca";
23     NavigableMap<String, String> turkey = new BinaryNavMap<>();
24     turkey.put(k: "uskudar", v: "istanbul");
25     turkey.put(k: "kadikoy", v: "istanbul");
26     turkey.put(k: "cekirge", v: "bursa");
27     turkey.put(k: "gebze", v: "koceali");
28     turkey.put(k: "niksar", v: "tokat");
29     turkey.put(k: "kecioren", v: "ankara");
30     turkey.put(k: "aksaray", v: "istanbul");
31     turkey.put(k: "foca", v: "izmir");
32     turkey.put(k: "manavgat", v: "antalya");
33     turkey.put(k: "kahta", v: "adivaman");
34     turkey.put(k: "biga", v: "canakkale");
35
36
37     System.out.println("The original set odds is " + turkey);
38     NavigableMap<String, String> m = turkey.subMap("uskudar", true, "gebze", false);
39     System.out.println("The ordered set m is " + m);
40     System.out.println("Set of navigable: " + turkey.entrySet());
41     System.out.println();
42     System.out.println("The first entry is " + turkey.firstEntry());
43     System.out.println("The first key is " + turkey.firstKey());
44     System.out.println();
45     System.out.println("The last entry is " + turkey.lastEntry());
46     System.out.println("The last key is " + turkey.lastKey());
47     System.out.println();
48     System.out.println("Ceiling entry of " + testing + " is: " + turkey.ceilingEntry(testing));
49     System.out.println("Ceiling key of " + testing + " is: " + turkey.ceilingKey(testing));
50     System.out.println();
51     System.out.println("Higher entry of " + testing + " is: " + turkey.higherEntry(testing));
52     System.out.println("Higher key of " + testing + " is: " + turkey.higherKey(testing));
53     System.out.println();
54     System.out.println("Floor entry of " + testing + " is: " + turkey.floorEntry(testing));
55     System.out.println("Floor key of " + testing + " is: " + turkey.floorKey(testing));
56     System.out.println();
57     System.out.println("Lower entry of " + testing + " is: " + turkey.lowerEntry(testing));
58     System.out.println("Lower key of " + testing + " is: " + turkey.lowerKey(testing));
59     System.out.println();
60     System.out.println("Tail map of " + testing + " is: " + turkey.tailMap(testing));
61     System.out.println("Head map of " + testing + " is: " + turkey.headMap(testing));
62     System.out.println();
63     System.out.println("Poll first: " + turkey.pollFirstEntry());
64     System.out.println("Poll last: " + turkey.pollLastEntry());
65
66

```

```

82
83 public static Boolean Q2Test()
84 {
85     HashMap<String, String> turkey = new HashTableChaining<>();
86     turkey.put("edremit", "balikesir");
87     turkey.put("edremit", "van");
88     turkey.put("kemalpasa", "bursa");
89     turkey.put("kemalpasa", "izmir");
90     turkey.put("ortakoy", "istanbul");//we assume a district
91     turkey.put("ortakoy", "aksaray");
92     turkey.put("ortakoy", "corum");
93     turkey.put("kecioren", "ankara");
94     turkey.put("pinarbasi", "kastamonu");
95     turkey.put("pinarbasi", "kayseri");
96     turkey.put("eregli", "konya");
97     turkey.put("eregli", "tekirdag");
98     turkey.put("eregli", "zonguldak");
99     turkey.put("golbasi", "adivaman");
100     turkey.put("golbasi", "ankara");
101     turkey.put("biga", "canakkale");
102
103     /* *test all
104
105         V get(Object key);
106
107         V put(K key, V value);
108
109         V remove(Object key);
110
111         int size();
112
113     * */
114     System.out.println(turkey.get("pinarbasi"));
115     System.out.println(turkey.put("baglarbasi", "istanbul"));
116     System.out.println(turkey.remove(key: "biga"));
117     System.out.println(turkey.size());
118     return Boolean.TRUE;
119 }
120

```