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

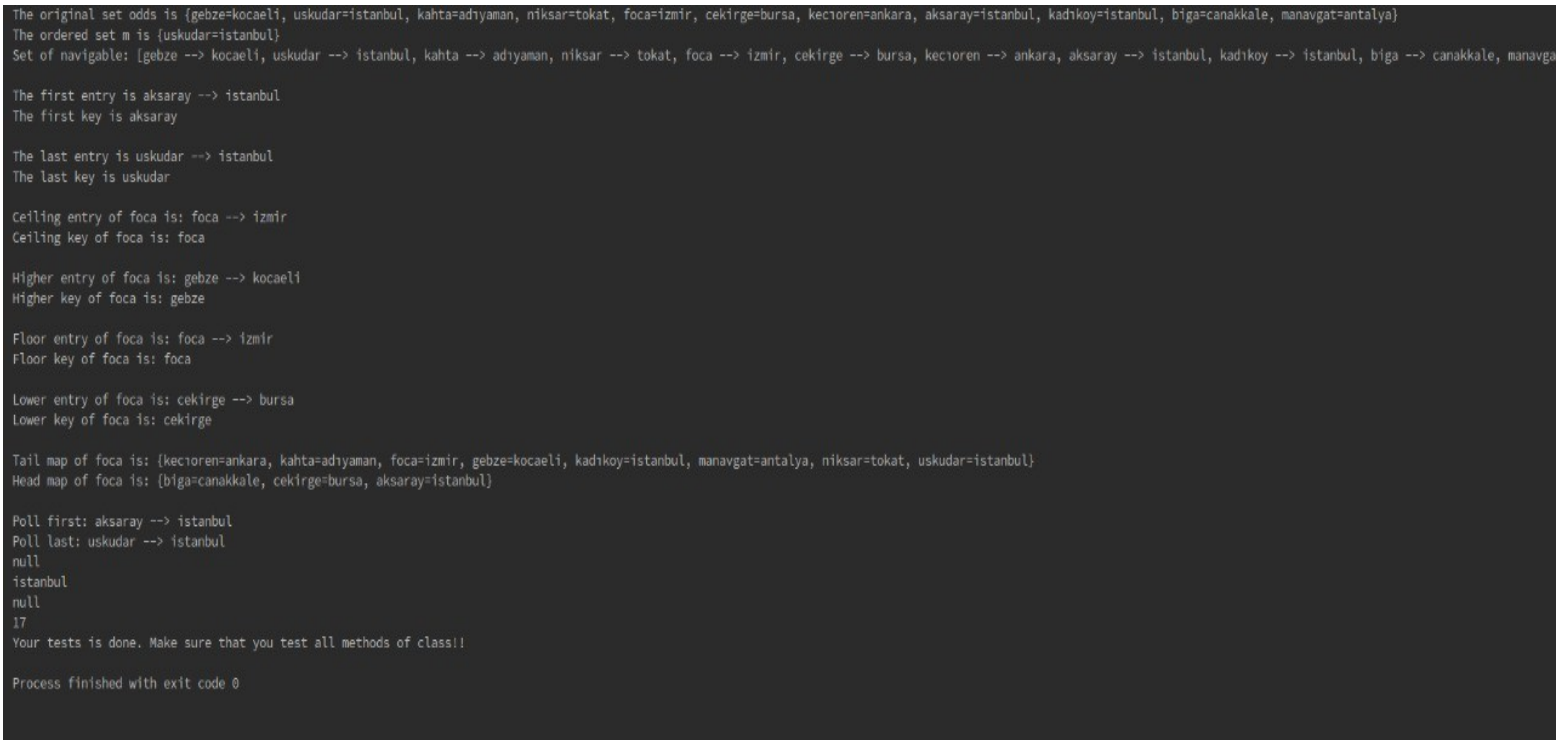
To build a navigable map a binary search tree has used as a container for BinaryNavMap. Although it is easy to use a binary search tree for insertion, extraction and search operations I found traversing the tree through an iterator much easier for this homework. In almost every method an treeIterator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



```java
public static Boolean Q1Test()
{
    String testing = "foca";
    NavigableMap<String, String> turkey = new BinaryNavMap<~>();
    turkey.put( k: "uskudar",  v: "istanbul");
    turkey.put( k: "kadıkoy",  v: "istanbul");
    turkey.put( k: "cekirge",  v: "bursa");
    turkey.put( k: "gebze",  v: "kocaeli");
    turkey.put( k: "niksar",  v: "tokat");
    turkey.put( k: "keçıoren",  v: "ankara");
    turkey.put( k: "aksaray",  v: "istanbul");
    turkey.put( k: "foca",  v: "izmir");
    turkey.put( k: "manaygat",  v: "antalya");
    turkey.put( k: "kahta",  v: "adıyaman");
    turkey.put( k: "biga",  v: "canakkale");



    System.out.println("The original set odds is " + turkey);
    NavigableMap<String, String> m = turkey.subMap("uskudar", true, "gebze", false);
    System.out.println("The ordered set m is " + m);
    System.out.println("Set of navigable: " + turkey.entrySet());
    System.out.println();
    System.out.println("The first entry is " + turkey.firstEntry());
    System.out.println("The first key is " + turkey.firstKey());
    System.out.println();
    System.out.println("The last entry is " + turkey.lastEntry());
    System.out.println("The last key is " + turkey.lastKey());
    System.out.println();
    System.out.println("Ceiling entry of "+ testing +" is: " + turkey.ceilingEntry(testing));
    System.out.println("Ceiling key of "+ testing +" is: " + turkey.ceilingKey(testing));
    System.out.println();
    System.out.println("Higher entry of "+ testing +" is: " + turkey.higherEntry(testing));
    System.out.println("Higher key of "+ testing +" is: " + turkey.higherKey(testing));
    System.out.println();
    System.out.println("Floor entry of "+ testing +" is: " + turkey.floorEntry(testing));
    System.out.println("Floor key of "+ testing +" is: " + turkey.floorKey(testing));
    System.out.println();
    System.out.println("Lower entry of "+ testing +" is: " + turkey.lowerEntry(testing));
    System.out.println("Lower key of "+ testing +" is: " + turkey.lowerKey(testing));
    System.out.println();
    System.out.println("Tail map of "+ testing +" is: " + turkey.tailMap(testing));
    System.out.println("Head map of "+ testing +" is: " + turkey.headMap(testing));
    System.out.println();
    System.out.println("Poll first: " + turkey.pollFirstEntry());
    System.out.println("Poll last: " + turkey.pollLastEntry());
```

```java
public static Boolean Q2Test()
{
    HashMap<String, String> turkey = new HashTableChaining<~>();
    turkey.put("edremit", "balikesir");
    turkey.put("edremit", "van");
    turkey.put("kemalpasa", "bursa");
    turkey.put("kemalpasa", "izmir");
    turkey.put("ortakoy", "istanbul");//we assume a district
    turkey.put("ortakoy", "aksaray");
    turkey.put("ortakoy", "corum");
    turkey.put("keçıoren", "ankara");
    turkey.put("pinarbasi", "kastamonu");
    turkey.put("pinarbasi", "kayseri");
    turkey.put("eregli", "konya");
    turkey.put("eregli", "tekirdag");
    turkey.put("eregli", "zonguldak");
    turkey.put("golbasi", "adıyaman");
    turkey.put("golbasi", "ankara");
    turkey.put("biga", "canakkale");

    /* *test all

        V get(Object key);

        V put(K key, V value);

        V remove(Object key);

        int size();

    * */
    System.out.println(turkey.get("pinarbasi"));
    System.out.println(turkey.put("baglarbasi", "istanbul"));
    System.out.println(turkey.remove( key: "biga"));
    System.out.println(turkey.size());
    return Boolean.TRUE;
}
```