

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK HW07 REPORT

Burak AKTEN
141044045

1. System Requirements

No needs for this homework.

2. Use Case Diagrams

No need for this homework.

3. Class Diagrams

No need for this homework.

4. Other Diagrams

For bonus part.

5. Problem Solutions Approach

• Question 1

- Bu soruda bizden NavigableMap interfacesini implement , AbstractMap clasınıda extend ederek BST kurallarına göre BinaryNavMap adında bir class implement etmemiz istendi.
- Bunun üzerine BST yapısının kullanılabilmesi için bu class içersine generik iki parametresi olan Node adında içersinde left ve right için olmak üzere iki tane Node referancısı tutan bir inner class implement edildi. Böylelikle oluşturulan map binary search tree olarak oluşturuldu.
- Ayrıca oluşturulan map'de bulunan Entry'lere erişebilmek içinde Map.Entry interfacesini ve Comparable interfacesini(key değğerleri arasında karşılaştırma yapabilmek için) implement eden MyEntry<K , V> adında diğer bir inner class implement edildi.
- Daha sonrasında bizden BinaryNavMap implement edilmesi gereken metotlar (implement edilen interfaceden ve extend edilen abstract class'dan gelen metotlar) implement edildi.
- Put(K , V) metodu için binary search tree algoritması uygulanarak map'a eklemeler gerçekleştirildi.
- Key 'ler için yapılan karşılaştırmalar için MyEntry inner classı içersinde implement edilen compareTo metodu kullanıldı.
- Oluşturulan Map tree üzerinde traverse işlemleri için genellikle Set Interfacesinde bulunan iteratör kullanıldı. Map içersinde iterator olmadığı için bir Entry<K ,V> seti oluşturularak o set üzerinde traverse gerçekleştirildi.
- Implement edilen entrySet metodu için return edilen set referansı map'ın level order traverse edilmesi sonucu elde edilen set referancısıdır.
- Implement edilmesi gerek çoğu metod için helper metodlar kullanıldı.

• Question 2

- Bu soruda bizden key değğerlerinin unique olmadığı ve bir HashMap imlement intmemiz istenildi. Oluşacak collision' ı azaltmak için içersinde Linkedlist yerine open adressing metodunu kullanan chaining metodu ile bu HashMap interfacesini implement etmemiz istenildi.
- Bunun üzerine chaining uqlamak için oluşturulan table'ın kapasitesi fix olacak şekilde 11 olarak ayarlandı. Oluşan table'ın her bir index'ide en az 11 elemalı olacak şekilde ayarlandı. Tabiki her bir eleman collision olucak olur ise bu sayı artabilir.
- Daha sonrasında oluşan bu table için her bir index null olarak set edildi ki collision olayını takip edebilelim.
- Yardımcı olarak 1.soru için kullanılan MyEntry adındaki inner class bu class içersinde de kullanıldı.
- HashMap interface'inde bulunan metotların implement edilmesinde uygulanan yöntemler ve çözüm yolları şu şekildedir:
 - **put(K ,V) :** Öncelikle verilen key için hashcode bulundu ve sonrasında bu kodun capasite ile modu alınarak entry'nin konulacağı index belirlendi. Daha sonrasında bulunan index 'de görülen ilk null kısma entry eklendi. Aynı key değğerine sahip birden fazla value olduğundan eklenen entry'nin value değeri döndürüldü.

- **get(Object key):** Bu fonksiyon iinde aynı işlemler yapılarak key için index değeri bulundu. Ve belirlenen index te bulunan array üzerinde gezilerek verilen key değeri için bulunan ilk value değeri döndürüldü. Bir key değeri için birden fazla value olduğundan dolayı ilk bulunan döndürüldü.
- **remove(Object key):** Aynı şekilde bu metod içersinde de verilen key değeri için bir index değeri hesaplandı. Daha sonrasında bu index deki array üzerinde gezilerek bu key değeri ile en son eklenen entry bulundu ve map'dan silindi.
- **size() ve isEmpty():** Bu metodlar için extra bir şey yapılmadı size değeri ile küçük işlemler yapıldı.

6. Test Cases

• Question 1

- ❖ Map ' a put işlemi gerçekleştirilmeden map objesinin toString() metodu çağırıldı ve beklenildiği gibi ekrana bir şey basmadı bunun üzerine eklemeler yapılarak aynı işlem gerçekleştirildiğinde obje için oluşturulan veri ekrana basıldı.
- ❖ Daha sonrasında toKey değeri fromKey değerinden küçük olan inputlar ile subMap metodu çağırıldı ve beklenildiği üzere IllegalArgumentException() fırlatıldı. Daha sonrasında uygun inputlar ile aynı metod tekrar çağırıldı ve doğru bir şekilde çalıştı.
- ❖ Map doldurulduktan sonra firstEntry() , lastEntry() , higherEntry ve lowerKey gibi metodlar kullanıldı ve beklenen sonuçlar elde edildi.
- ❖ Bu işlemlerden sonra mapdan pollLastEntry() ve pollFirstEntry() metodları ile silme işlemleri gerçekleştirildi ve silinen Entry değerleri beklenildiği üzere return edildi.
- ❖ Oluşturulan objenin descendingMap haline getirmek için descendingMap metodu çağırıldı ve beklenildiği gibi map reverse edildi.
- ❖ navigableKeySet ve descendingKeySet metodları çağırıldı ve elde edilen sonuçlar ikisininde implementasyonlarında TreeSet kullanıldığı için aynı oluştu.

• Question 2

- ❖ Oluşturulan HashMap objesine ekleme işlemleri gerçekleştirilmeden isEmpty() metodu çağırıldı ve beklenen gibi false return edildi. Daha sonra ekleme işlemleri gerçekleştirildikten sonra aynı metod tekrardan çağırıldı ve true return etti.
- ❖ Ekleme işlemi gerçekleştikten sonra get() metodu mapda olmaya bir key için çağırıldı ve beklenildiği gibi exception fırlatıldı. Sonrasında bir değer alabilmek için aynı metod var olan bir key ile kullanıldı ve implementasyona göre bir değer return edildi.
- ❖ Daha sonrasında var olan bir key ve değeri silinmesi için remove metodu kullanıldı ve sonrasında size metodu ile beklenildiği gibi size'ın azaldığı ve silme işleminin gerçekleştiği görüldü.
- ❖ En sonunda ise toString metodu ile oluşturulan objenin içeriği ekrana bastırıldı.

7. Running and Results

- Question 1

```
The ordered set m is
niksar->tokat
kahta->adiyaman
manavgat->antalya
kadikoy->istanbul
kecioren->ankara
uskudar->istanbul

The first entry is aksaray->istanbul
The first key aksaray
The last entry is uskudar->istanbul
The last key uskudar
The lower entry of uskudar kadikoy->istanbul

The higher key of kahta kadikoy
pollFirstEntry() returned : aksaray->istanbul

pollLastEntry() returned : uskudar->istanbul

The navigableKeySet of the object :
[biga, cekirge, foca, gebze, kadikoy, kahta, kecioren, manavgat, niksar]
The descendingMap of the object :
niksar->tokat
kahta->adiyaman
kecioren->ankara
biga->canakkale
cekirge->bursa
kadikoy->istanbul
gebze->kocaeli
manavgat->antalya
foca->izmir

The descendingKeySet of the object :
[biga, cekirge, foca, gebze, kadikoy, kahta, kecioren, manavgat, niksar]
```

Run - _hw7

```
Run Main
*****Q1 TESTING*****

The original set odds is
niksar->tokat
aksaray->istanbul
uskudar->istanbul
keciooren->ankara
manavgat->antalya
cekirge->bursa
gebze->kocaeli
foca->izmir
kadikoy->istanbul
biga->canakkale
kahta->adiyaman

The ordered set m is
manavgat->antalya
niksar->tokat
kahta->adiyaman
kadikoy->istanbul
keciooren->ankara
uskudar->istanbul

The first entry is aksaray->istanbul

The last entry is uskudar->istanbul
The last key uskudar
The lower entry of uskudar kadikoy->istanbul

The higher key of kahta kadikoy
pollFirstEntry() returned : aksaray->istanbul

The removed(pollFirstEntry()) set odds is
keciooren->ankara
kahta->adiyaman
niksar->tokat
kadikoy->istanbul
biga->canakkale
foca->izmir
gebze->kocaeli
manavgat->antalya
uskudar->istanbul
```

- **Question 2**

```
*****Q2 TESTING*****

The original set odds is
[biga->canakkale, null, null, null, null, null, null, null, null, null, null]
[golbasi->adiyaman, golbasi->ankara, null, null, null, null, null, null, null, null, null]
[null, null, null, null, null, null, null, null, null, null, null]
[null, null, null, null, null, null, null, null, null, null, null]
[null, null, null, null, null, null, null, null, null, null, null]
[keciooren->ankara, eregli->konya, eregli->tekirdag, eregli->zonguldak, null, null, null, null, null, null, null]
[kemalpasa->bursa, kemalpasa->izmir, null, null, null, null, null, null, null, null, null]
[ortakoy->istanbul, ortakoy->aksaray, ortakoy->corum, null, null, null, null, null, null, null, null]
[pinarbasi->kastamonu, pinarbasi->kayseri, null, null, null, null, null, null, null, null, null]
[edremit->balikesir, edremit->van, null, null, null, null, null, null, null, null, null]
[null, null, null, null, null, null, null, null, null, null, null]

The hashmap is empty? : false
The size of hashmap : 16
The value of golbasi : adiyaman
The value of remove(biga) : canakkale
The value of remove(golbasi) : ankara
The size of hashmap : 14
Your tests is done. Make sure that you test all methods of class!!

Process finished with exit code 0
```

