

MUZAFFER METEHAN ALAN –

151044038

CSE222 - SPRING 2017

HW7

Genel Bilgiler :

- Öncelikle ilk soru için , daha önceki ödevlerde implement ettiğimiz BinaryTree ve BinaryTreeSearch classlarını projeye ekledim.
- Daha sonra composition olarak BinarySearchTree objesi oluşturdum.
- BinaryNavMap classındaki fonksiyonları bu BST kurallarına uyarak implement ettim.
- Fonksiyonları main classında test ettim ve sonuçların ekran görüntüsünü aldım.
- 2. Soru için chaining sağlanması için LinkedList array objesi oluşturdum.
- Put,remove,get,size fonksiyonlarını implement ettim.
- Fonksiyonları main classında test ettim ve sonuçların ekran görüntüsünü aldım.
- Javadoc tamamlanıp eklenmiştir.

Test Case

1.soru için;

Put() fonksiyonu Key,Value içeren objeleri parametre olarak alır ve BST ye kurallar doğrultusunda ekleme yapar.Null olarak eklenirse bir şey eklenmez.

EntrySet() fonksiyonu BST'den bir set oluşturarak bunu return eder, eğer BST boş ise boş bir set return edilir.

LowerEntry() fonksiyonu verilen Key'den daha compare'a göre daha düşük olan en yüksek Entry'i return eder.Eğer bulamazsa null return edilir.

LowerKey() fonksiyonu , lowerEntry() fonksiyonunda döndürülecek olan Entry'nin Key'ini return eder.Yoksa null return edilir.

FloorEntry() fonksiyonu, verilen Key'den daha compare'a göre daha düşük olan en yüksek Entry'i return eder.Eğer bulamazsa null return edilir.

FloorKey() fonksiyonu, verilen Key'den daha compare'a göre daha düşük olan en yüksek Entry'i return eder.Eğer bulamazsa null return edilir.

CeilingEntry() fonksiyonu , verilen Key'den daha büyük veya kendisine eşit olan en küçük entry'i return eder.Eğer bulamazsa null return edilir.

CeilingKey() fonksiyonu , verilen Key'den daha büyük veya kendisine eşit olan en küçük Key'i return eder.Eğer bulamazsa null return edilir.

HigherEntry() fonksiyonu , verilen Key'den daha büyük veya kendisine eşit olan en küçük entry'i return eder.Eğer bulamazsa null return edilir.

HigherKey() fonksiyonu , verilen Key'den daha büyük veya kendisine eşit olan en küçük Key'i return eder.Eğer bulamazsa null return edilir.

FirstEntry() fonksiyonu , en düşük değerli Key'in entrysini return eder.Yoksa null return eder.

LastEntry() fonksiyonu , en yüksek değerli Key'in entrysini return eder.Yoksa null return eder.

FirstKey() fonksiyonu en düşük değerli Key'ini return eder.Yoksa null return eder.

LastKey() fonksiyonu en yüksek değerli Key'ini return eder.Yoksa null return eder.

PollFirstEntry() fonksiyonu en düşük değerli Key'i BST den siler ve return eder.Yoksa null return eder.

PollLastEntry() fonksiyonu en yüksek değerli Key'i BST den siler ve return eder.Yoksa null return eder.

DescendingMap() fonksiyonu BST'yi azalacak şekilde yani tersten sırasıyla olacak şekilde bir NavigableMap return eder.Yoksa null return eder.

DescendingKeySet() fonksiyonu BST'yi azalacak şekilde yani tersten sırasıyla olacak şekilde bir KeySet return eder.Yoksa null return eder.

NavigableKeySet() fonksiyonu BST'yi normal artar şekilde yani normal sırasıyla olacak şekilde KeySet return eder.Yoksa null return eder.

SubMap() fonksiyonu kendisine verilen ilk parametreden 3. Parametreye kadar bir sub Map oluşturur.Verilen boolean parametrelere göre de bu Keyler dahil olur.Bulunamazsa null return edilir.

Problem Solution Approach

Öncelikle Map fonksiyonlarının nasıl çalıştığına dair analizleri yaptım ve bu fonksiyonları BST'ye göre nasıl implement edeceğimi belirledim.Fonksiyonlarda compare'a göre düşük valueların root'a göre solda , büyük olanların sağda olduğu mantığıyla çalışıyor.

Chaining classı içinse yapılacak olan eklemeleri aynı olan Keyler için aynı node'da birden fazla tutulacak şekilde implement ettim.

