

MUZAFFER METEHAN ALAN –

151044038

CSE222 - SPRING 2017

HW6

Genel Bilgiler:

- Öncelikle verilen soruların gerekli analizlerini yaptım.
- Daha sonra kitapta ilgili alanların konularını okudum ve kitaptaki implementasyonları anlamaya çalıştım.
- İlk soru için complete tree oluşturmayı (Array) yardımıyla implement edebildim fakat bizden nodeları kullanarak yapmamız isteniyordu. Bu şekilde implementasyonunu yapamadığım için 1.soruyu gönderemedim.
- Daha sonra 2.soru için kitapta bulunan HuffmanTree classını olduğu gibi aldım. Kodun yapabildikleri , binaryTree classından aldığı fonksiyonlar, verilen bir HuffData arrayinden HuffmanTree build edebilmek ve decoding yapabilmektir. Bizden istenilen encoding fonksiyonu yazmamız ve main classında decoding ve encoding fonksiyonlarını test etmemizdi.
- Decoding fonksiyonu kendisine verilen encrypted bir string i örneğin ("1011)' in ağaçta gittiği path doğrultusunda gösterdiği karakteri döndürüyordu.
- Encoding fonksiyonunun ise bu işin tam tersini yapacak şekilde kendisine verilen bir stringi encrypt etmesi gerektiğini düşündüm.
- Yani buildTree fonksiyonu ile build edilen ağaçtan elde edilen A harfi için = "1001" , B harfi için = "0101" , C harfi için = "101" şifrelerini fonksiyona ("ABC") verdiğimizde = "10010101101" return edecek şekilde implement ettim. Bu iki fonksiyonu main classında teker teker test ettim. Dosyadan okunacak olan tüm karakterler öncelikle decode ardından encode işlemlerine tutulur. Daha sonra

hepsinin birleşimi sonucunda oluşan String encoding edilir.Yaptığım test sonuçlarını screenshot olarak ekledim.

- *3.Soru için 5. ödevde implement ettiğimiz FamilyTree classını olduğu gibi alıp sadece levelOrderTraverse fonksiyonunu ekledim.Daha sonra yine geçen ödevde yollanmış olan main classında bu levelOrderTravers fonksiyonunu çağırdım, test ettim.*
- *Test sonuçlarını valid ve invalid inputlarla birlikte screenshot olarak ekledim.*
- *2. Ve 3. Sorular için Class Diagram ve Javadoclar tam olarak yapıldı.*

Problem Solution Approach

3.soru için kitapta bulunan , family tree algoritmasını kafamda tasarladım ve bu algoritmaya uygun insert methodunu implement ettim.Eğer eklenecek olan kişinin parent ı ve nick 'i aynıyken , aynı kişi eklenmeye çalışılırsa exception fırlatılmasını gerçekleştirdim.Onun dışında yeni olarak eklenen her child , eğer ilk child ekleniyorsa parentın sağına daha sonra eklenecek olan childlar ilk eklenen childın soluna eklendi. Daha sonra levelOrderTraverse fonksiyonunu depth(derinlik fonksiyonu) kullanarak recursive olarak implement ettim.Bunu yapma sebebim levelOrder traversin basamak basamak ilerlemesindendi.

Daha sonra FamilyTree classını test edecek olan main classını implement ettim , pdfte verilen testi uyguladım.

2.soru için için kitapta bulunan , huffman tree algoritmasını olduğu gibi aldım , öncelikle kodun neler yapıp neler yapamadığının analizini yaptım.Kod için herhangi bir ekleme fonksiyonu gerekmiyordu buildTree fonksiyonu sayesinde huffData arrayi verilerek ağaç build edilebiliyordu.

Daha sonra encoding işleminin HuffmanTree için ne anlama geldiğinin araştırmasını yaptım pek bir açıklama bulamadım ama decoding işleminin yaptığı iş açısından tam tersini yapar şekilde çalışması gerektiğini düşündüm.Encoding fonksiyonunu 2 farklı fonksiyon kullanarak , birisi public ve kullanıcı onu çağırıyor diğer ise private ve bu public fonksiyon

tarafından çağırılıyor.Recursive olarak çalışan bu fonksiyon verilen String i tek tek parçalarına ayırıp o karakteri arıyor.Ararken geçtiği path'i recursive olarak sola giderse "0" , sağa giderse "1" ekleyerek ilerliyor.En son ise verilen Stringin encoded halini return ediyor.

Test Case

3.soru için test sonuçlarını screenshot olarak ekledim.

Eğer fonksiyonlar valid şekilde kullanılırsa herhangi bir exception fırlatılmaz ve istenilen sonuç ekrana bastırılır,ağaca eklemeler yapılır.

Eğer verilen parent , ağaçta bulunamazsa ekleme yapılırken exception fırlatılır.Ekrana "Böyle bir parent bulunamadı" mesajı bastırılır.

Eğer bir parent'a (aynı nickname için) aynı child eklenmeye çalışılırsa (pdf'te belirtilen kontrol edilmesi gereken durum) exception fırlatılır.Bu exception ekrana "Bu parent'a tekrardan bu kişiyi ekleyemezsiniz" mesajı bastırılır.

2.soru için test sonuçlarını screenshot olarak ekledim.

Eğer fonksiyonlar valid şekilde kullanılırsa herhangi bir exception fırlatılmaz , decoding ve encoding fonksiyonları yapması gerektiği şeyleri yapar.

Eğer buildTree fonksiyonu kullanılmadan herhangi bir fonksiyon kullanılmaya çalışılırsa , tree tanımlanması yapılmadığı yani içi doldurulmadığı için exception fırlatır.