

Gebze Technical University  
Computer Engineering

CSE 222  
2017 Spring

HOMEWORK 5 REPORT

Furkan AKTAŞ  
141044029

Course Assistant:

## Problem Solutions Approach

### Q1) Part1-)

Öncelikle kitaptaki node yapısı vs. gerekli veriler alındı ve BinaryTree oluşturuldu.Oradaki readBinaryTree fonksiyonu bizim istediğimiz yapıda tekrar yazıldı.Iterator fonksiyonu yazıldı.Bu Iterator fonk için TreeIter class'ı yazıldı.Bu iterator, pre order traverser sistemine uygun olarak işlem yapmaktadır.Bu işlemleri class içinde bulunda stack yapısıyla sağlandı.İlk işlem olarak constructordan gelen root stack'e atıldı.Sonra her next() fonksiyonu çağrıldığında, stack 'teki eleman peek edildi(veriyi tempde tutarak ekrana basabilmek için), sonra pop edildi.Daha sonra önce null değilse sağ ağaç, sonra sol ağaç stack'e eklendi(çıkartırken önce sol çıksın diye). Bir sonraki next() çalıştığında önce sol ağaç çıkmış olacak ve o çıkan sol ağacın sağ ve solu root ları stack'e eklenmiş olacak.

### Part2-)

Öncelikle BinarySearchTree , BinaryTree' den extend edildi.BinaryTree den farklı olarak Buna yeni iterator sınıfı yazıldı.Bu class(levelOrderIterator), level order traverser sistemine uygun olarak yazıldı.Bu class tada , node alan parametresi olan constructor bulunmaktadır.Level order sistemini uygulayabilmek için bunda Queue kullanıldı.Constructor dan gelen node , queue nun ilk elemanı olarak eklendi. Next() fonksiyonunda , önce peek() sonra poll() yapıldı, peşinden çıkarılan node'un önce left'i sonra right node'u queue ya eklendi.Bir sonraki next()'te önce eklenen sol çıkarıldı sonra o çıkarılanın solu ve sağı sona eklendi , daha sonra sağ node çıkarıldı sonra o node un solu sağı sona eklendi bu sayede travers edildi.Queue kullanarak aynı seviyelerdeki verileri basarak travers etmiş olduk.

### Q2)

Öncelikle BinarySearchTree 'den FamilyTree'yi extend ettik.Eğer constructor dan parameter olarak gelen isim dosyanın ilk ismiyle uyuşmazsa program çalışmaz. Önce dosyadan satır okunur.Sonra satır “,” 'e göre split edilir.İlk ikisi name ve parent değişkenine verir. 3. sü “-” 'e göre split edilir. İlki nick'e ikincisi nickname'e atılır.Eğer nick Ebu ise name parent aynı sistemler olurken , Ibn ise name ve parent ters kaydedilir bunun sebebi her zaman önce üst node'u arayabilmek.

Search fonksiyonunda önce parent bulunur.Sonra, eğer nick ebu ise önce sol node a bakıldı eğer boşsa ve name ve nickname aynı ise (Ayşe, hasan, ebu-ayşe) oraya eklendi.Eğer sol boş değilse sol node goRight fonksiyonuna verildi.Burada eklenen kişi ile gelen node aynı olmadığı sürece sağ node null olana kadar ilerlendi ve en son sağa eklendi.

Eğer nick Ibn ise, parent ın leftine bakıldı null değilse ibnFinder fonksiyonuna left gönderildi.Burada eğer bu node eklencek verinin parent'ı ise node'un sol tarafına ekler değilse, root right ı ile tekrar ibnFinder çağrıldı. Root null olana kadar bu devam etti(null ise sonlandı ).

Eşleşme olsada olmasada tüm ağaç gezilir.Her eşleşmede private count değişkeni değeri 1 artırılır.En son count kontrol edilir, count 1 den çoksa aynı parent ve lakapta birden fazla veri var.Eğer 0 'sa eşleşme bulunamamış demektir.

## Class Diagrams







