

VERİ YAPILARI ÖDEV RAPORU

İstenilen: Veri yapıları dersinde bugüne kadar öğrenilen birçok uygulamanın kullanıldığı bir test ödevi. Bir txt dosyasındaki sayılar okunarak istenilen liste, ağaç methotlarıyla bir takım işlemler yapılarak ekrana çıktı alınması bekleniyor.

Öğrendiklerim: Ödevi yaparken c++ syntax'ına daha çok hakim oldum diyebilirim. AVL ağacının, radix sortun çalışma mantıklarını da öğrendim. Ayrıca yazdığımız kod methodlarına göre performans hızının ne kadar değiştiğine de şahit oldum.

Yaptıklarım: Öncelikle hücre ve dokuyu bir satır listesi-yönetici listesi mantığında hiyerarşik olarak düşündüm. Bunun için de önceki ödevdeki bağlı liste mantığını her ikisi için de kullandım ve hücreyi dokunun içerisinde bir data olarak tuttum, yani dokularımızda hücrelerimiz yer alacak. Hücrelerimiz de veri.txt'deki sayıları tutan(bunları dna uzunlukları olarak adlandırdım) bir çift yönlü listelerdir. Dokuları, dosya okuyucu sınıfından tüm hücreleri sayılarla oluşturarak daha sonra bu hücrelerle mainde dolduruyoruz. Daha sonrasında organı da bir ağaç olarak düşünerek, BST class'ının kullanacağı node yapısını da ayrıca OrganNode şeklinde isimlendirdim. Main kısmımızda bir for döngüsüne giriyoruz ve tüm dokuları okutuyoruz. Okuttuğumuz dokuyu bir diziye alarak radix sort içerisinde sıralatıyoruz ve orta elemanını o anki organ(bst)'nin içine ekliyoruz. Her 20 dokudan oluşan sayılık organda bir AVL denge kontrolünü organların üzerinde uygulatarak o ağaç dengeli mi değil mi diye bakıyoruz, eğer dengeliyse # karakterini değilse boş karakter yazarak program geçiyor. Bu kontrol sonrasında organı sıfırlayarak diğer 20'lik organa geçiyoruz. Sonuç olarak tüm sayıları dolaştığımız bir for loop mantığında düşündüm. Mutasyona gelirse de yine benzer bir mantık var ancak ekstra olarak, 20'lik organ oluştuğunda öncelikle köküne firstofroot fonksiyonuyla baktığımız bir 50'ye bölünüyor mu kontrolü var. Bölünüyorsa her bir dokudaki hücrelerin dnalarını ayrı ayrı dolaşıp 2'ye bölünüyorsa 2'ye bölerek mutasyona uğratıyoruz. Daha sonrasında bu yeni sayılarımızı aynı mantıkta sıralatıp, sıfırlamış olduğumuz organa ekleyerek yine aynı şekilde kontrollerle çıktı alıyoruz.

Yapamadıklarım: Ödevde bir sistem ve organizma sınıfı mantığına ihtiyaç duymadım, uğraşmış olsam da yapamadım. Aslında sistemin içinde 100 tane organ dizisi bulunacak ve bunların içlerini doldurup kontrol edeceğiz şeklinde düşünmüştüm ancak ne yazık ki run errorü aldım. Bunun bellek taşmasıyla ilgili olabileceğini düşünüyorum ama gerekli vaktim kalmadığımdan artık deneyemeyeceğim.

Aşağıda programın çalışma çıktılarına dair görüntülere ulaşabilirsiniz. Program adım adım yazdırarak gidiyor.

```
ORGANIZMA(NORMAL)

#
#####
#####
#####
#####
#####          #
#### #####    ##
#####        ###
#####
#####
#####
#####        ##
#####        #
#####
#####
#####
#
```

Mutasyona ugratmak için rastgele bir karaktere basıp enterlayabilirsiniz.

```

ORGANIZMA (MUTASYONA UGRADI)

#           #   #
#           #####   #
#           #   #####   #           #           #
#           #####   #   #           #
#   #####   #####   #           #           #
#   #####   #####   ##           #           #
#####   #####   ##           #
#####   #####   #           #   #
#####   #####   ##           #
#####   #####   #           #
#####   #####   ##           #
#####   #####   #           #
#####   #####   #   #   #   #
#####
#####
#####   #   #   ##           #           #
##           #   #   ##           #
C:\Users\Berkay\Desktop\dokunuImayacak\dogru mutasyo\odevorganizma>

```