

Ad: Yusuf İslam

Soyad: Bil

Okul Numarası: B201210002

Grubu: 1-C

Ders Adı: Veri Yapıları

Ödev Numarası: 2. Ödev

Okul Adı: Sakarya Üniversitesi

Bölüm: Bilgisayar Mühendisliği

Ödev Tanıtımı

Program dosyalarım 17 class dosyasından ve bunların kütüphane dosyalarından oluşuyor. Programın çalıştığı ana dosyam main.cpp dir. Programın amacı bir veri.txt dosyasındaki sayıları okuyup hücre doku organ sistem organizma olacak şekilde bir zincir mantığıyla birbirine bağlamak. Dosyamızı okurken bu döngünün içinde bütün bağlantılarımızı yapıyoruz. İlk satırımızı okuduktan sonra radix sort ile küçükten büyüğe sıralıyoruz daha sonra bu sıralanmış verileri hücre bağlı listemize ekliyoruz ve orta sıradaki değeride doku bağlı listemize ekliyoruz. Her satır bu mantık ile birbirne bağlanıp ortanca sayıda doku listesine bağlanıyor. Her 20 doku oluştuğunda bir organ yapıyoruz ve doku listemizin ilk elemanını organ listemizin içinde bulunan ikili arama ağacı listesine kok olarak ekliyoruz. Organ yapısına katılacak diğer dokularıda kok yapısına ekle fonksiyonuyla ekliyoruz. Eğer eklenecek sayı kok değerinden küçük eşit ise sol tarafa eğer büyük ise sağ tarafa ekleme yapıyoruz. Daha sonra 100 organ bir sistem oluşturacak şekilde koşullarımızı koyup her organın ilk kok değerini sistem bağlı listemize ekliyoruz. Böylelikle adres bilgilerimizi kaybetmeyip aşağıdan yukarı çıkabildiğimiz gibi yukarıdan aşağıyada inebiliyoruz. 100 organ bir sistemi oluşturunca bu sistem bağlı listesinin ilk elemanını Organizma bağlı listesine ekliyoruz. Yani her 2000 satırlık veri bir sisteme denk geliyor ve veri.txt dosyasındaki satır sayısının miktarına göre organizmanın içindeki sistem sayısı oluşuyor, bunun bir sınırı yok. Ekran yazdırma kısmına geldiğimizde sistemimizi oluşturan organlara gidip, bu organ yapısındaki ağaç kok değerine gidip bu ağacın avl ağaç mantığına göre dengeli yada dengesiz olup olmadığını kontrol ediyoruz. Eğer ağacın iki kolu arasındaki yükseklik farkı 2 ve 2 den büyükse dengesiz, 2 den küçükse dengeli bir yapıya sahip. Bu ağaçların dengeli olanları için ekrana boşluk karakterini koyup, dengesiz olanlar için '#' karakterini ekrana yazdırıyoruz. Her sistemi 100 organ oluşturduğu için 1. Sistemin kontrolü bittiğinde bir alt satıra geçiyoruz ve diğer sistemin organlarını kontrol ediyoruz. Organizmayı ekrana yazdırdıktan sonra klavyeden enter tuşuna basıldığında ekrana çıkan organizmayı ekrandan temizleyip yerine mutasyona uğramış organizmanın çıktısını yazdırıyoruz. Mutasyonun oluşma koşuluda şu şekilde oluşuyor. Sistemimizde bulunan ikili arama ağaçlarındaki kök yapısına gidip bu kökün adresinde tuttuğu veri değeri eğer 50 sayısına tam bölünebiliyorsa bu ağaç yapısı mutasyona uğruyor, eğer bölünmüyorsa o ağaç mutasyona uğramıyor. Mutasyona uğrayacak ağaçlar içinde bu ağacı oluşturan her bir dala yani her bir dokuya gidip bu dokuyu oluşturan hücrelerin dna sayısına bakıyoruz yani adresteki veri sayısına. Bu veri değeri çift sayı ise 2 ye bölerek kaydediyoruz eğer çift bir sayı değilse olduğu gibi bırakıyoruz. Bu işlemi yapmak içinde ağaç yapısını post order sıralama metodu ile bir diziye atıyoruz ve oradaki sırayla bir dosyadan okurmuş gibi yeni

sırasına göre tekrardan mutasyon koşullarını yapıp ağaç yapısına ekliyoruz. Bu işlemler sonucu tekrar bütün sistemlerin içindeki ağaç yapısına gidip bir kez daha avl ağaç mantığına göre dengelimi dengesiz mi diye kontrol edip yine dengeli ise boşluk dengesiz ise '#' işaretini koyuyoruz ve ekrana tekrardan mutasyonlu organizma halini yazdırıyoruz. Programın çalışma şekli ve mantığı bu şekildedir.

Ödevde Zorlandıklarım

Veri.txt dosyası çok uzun olduğu için ve iç içe çok fazla yapı kullandığımız için bu yapıları sıralarken karışıklıktan dolayı zorlandım. Ayrıca mutasyon kısmında ağaç yapısının içlerine girip güncelleme kısmında zorlandım.

Ödevden Öğrendiklerim

C++ dilinin diğer dillere göre çok zor olduğu ve en ufak bir mantıksal hataya yol açmamamız gerektiği. İçerik olarak radix sort ile kuyruk yapısını, ikili arama ağacını ve bağlı liste yapısını bir arada kullanmayı öğrendim. Nesneden diğer nesneye geçiş yaparak bir üst kısma geçiş yapmayı öğrendim. Ayrıca Avl ağaç dengesini ve post order sıralamasıyla koşullarımı kullanmayı öğrendim.