

RAPOR

Başlangıçta genel olarak, standart yapılar olarak görebileceğimiz düğüm oluşturma, düğüme eleman ekleme, verilen değerin düğümdeki yerini bulma gibi kısımlardan bahsetmek istiyorum. Bu fonksiyonları çoğunlukla sizlerin derslerinden öğrenmiş olduğum ve kitabınızı referans alarak hazırladığım kodlar ile kolaylıkla programıma eklemiş oldum. Bu kısımlarda bir problem çıkmamıştır. Bunlara ek olarak “dataOf” kısmını ekleyerek, istenen indexteki elemana ulaştım.

Standart olarak gördüğüm kısımlardan kısaca bahsettikten sonra programımın daha özgün kısımlarına gelecek olursak, en başta “algoritma” dosyasına eklediğim “int Obeg()”fonksiyonunu anlatmak istiyorum. “algoritma.cpp” dosyasının 30-55. Satırları arasında yer alıyor. Programımın çalışma prensibinde öncelikle txt dosyasından okumuş olduğum veriyi bu fonksiyona gönderiyorum. Sonrasında fonksiyon içerisinde bu eleman ile düğümde bulunan elemanları en baştan başlayarak OBEB değerlerini hesaplıyor. Yine bilinen ve kullanılan basit bir obeg hesaplama yöntemi ile hesapladım. Sonrasında bu obeg değerini daha önceden kayıt ettiğim en büyük obeg değeri ile karşılaştırıyorum ve algoritmada belirtildiği üzere karşılaştırma sonucunda, düğümdeki hangi değerde koşul sağlanıyorsa bu değeri return ederek fonksiyonu sonlandırmış oluyorum. Yani algoritmada belirtilen OBEB ile ilgili algoritmayı bu kısımda sağlamış oluyorum.

Obeg algoritmasından sonra elde etmiş olduğum değeri(Obeg fonksiyonundan return edilen değer) main fonksiyonu içerisinde kontrol ediyorum. Bu kontrol “calistir.cpp” dosyasının 35-58. Satırları arasına gerçekleşiyor. Bu kısımda MOD alma işlemine göre sayının yerleştireceği düğümü belirliyorum. Bu kontrol ilk olarak “Sayilar.txt” dosyasından okuduğumuz veri ile Obeg fonksiyonundan gelen verinin hangisinin daha büyük olduğunu belirlemekle başlıyor. Daha sonra karşımıza 3 adet kontrol çıkıyor. Birincisi eğer mod=0 ise. 2. Ve 3. Durumlar için bir “i” değişkeni tanımladım ve bu değişken 0 değeri ile başlıyor ve her yeni veri okunduğunda bir artıyor. Buradaki amacım kaçınıcı indexte olduğumu bilmek ve algoritmada belirtilen Mod alma işlemi sonucuna göre verilerin yerleştireceği konumu kolaylıkla belirlemek. 40. Satırda görüldüğü üzere i değişkenim index değerim ve mod değerim i değerimden büyükse yani algoritmadan belirtilen “Mod değeri kadar sola ilerle, eğer listenin başına gelmişsen en başa ekle” bölümünün karşılığını bizlere veriyor. Yani benim Mod değerim index değerimden büyük ise listenin başına ekle. 3. Kontrolüm ise algoritmada verilen son koşul olan index<mod kontrolünü sağlıyor. Bu üç kontrol sonucunda gerekli index belirlenerek txt dosyasından okuduğumuz sayının eklenmesi işlemi yapılıyor. 36. Ve 47. Satırlardan sonra gelen iki işlem bu şekildedir. 36. Ve 47. Satırlarda ise yalnızca iki sayıdan hangisinin büyük olduğunu belirlemiş olduk.

60. satır ile başlayan kontrol. Programın kilit noktalarından en önemlisi. Burada, yukarıda anlatmış olduğum i değerinin sutun değerine denk olup olmadığını kontrol ediyoruz. Bunun sebebi ise txt dosyası içerisinde bir satırın sonuna gelindiğini anlamak. Sutun sayısı programın çalışma sayfasında belirttiğim üzere “7” olarak girilmelidir. Programın doğruluğunun kontrolü adına txt dosyası içerisine her sütunda 7 eleman olan sayılar tanımladım. Sutun değerim programın başlangıcında kullanıcı tarafından verilmelidir. 60. Satırdaki kontrolün içerisine girildiğinde bir satırın sonuna geldik ve burada yeni bir dizi karşımıza çıkıyor. **Bir düğüm üzerinden direkt olarak sayıların ASCII karşılıklarını bulma**

RAPOR

konusunda eksik kaldım. Bu eksiği gidermek adına ve ayrıca **Dizi yapısını daha iyi anlayabilmek adına** “cevir” bir dizi tanımladım. Bu dizi tanımlama amacım, Düğümde bulunan verileri bu dizi içerisine göndererek (int değer olarak) her değerın ASCII karşılığını ekrana yazdırmak. Dizi içerisine gönderme işlemi, algoritma sonucu sayıların sıralanmış hali şeklindedir. Dizi üzerinde tekrar bir sıralama işlemi yapmamaktayız. Dizinin tek amacı verilerin üzerine kaydedilmesi ve ascii karşılıklarının ekrana yazılmasıdır. Sonrasında aynı if bloğu içerisinde dizinin her elemanını siliyorum. Aynı şekilde “sayılar” düğümünün de tüm düğümlerini temizliyorum. Bu temizleme işlemleri yalnızca dizi ve düğümün içerisindeki elemanların silinmesini hedeflemektedir. Böylece her satırda yeniden bir düğüm ve dizi tanımlamak zorunda kalmadan var olan dizi ve düğümü kullanıyoruz. Ayrıca i değerini tekrar 0’a eşitleyerek yeni bir satıra geçtiğimizi anlamış oluyoruz.

Son olarak main metodunun bitişinde çöp oluşmasını engellemek adına dizimizi ve düğüm yapımızı tamamen silmiş oluyoruz.

Ödevin yapım aşamasında oldukça fazla yeni bilgiler edindim. “Kod yazmadan öğrenilmez” cümlesinin doğruluğunu tekrar ispatlamış oldum. Bunların en başında, bilgi açısından eksik olduğum “pointer” konusu yer almakta. Kaynak kod ve başlık dosyalarına ayırma konusunda bilgilerimi pekiştirdim. Genel olarak döngüler ve if kontrolleri noktasında bir problem yaşamadım. Bu ödev sayesinde, bundan sonra “Liste Veri Yapıları” konusunda oldukça rahat kodlar yazacağıma inanıyorum.

Zorlandığım en büyük noktalardan biri internet ve bilgisayar erişimi kısmı oldu. Bu noktada zorluklar yaşıyorum ve buna bağlı olarak kısa sürede ödevi tamamlamak zorunda kaldım. Kodlama aşamasında ise düğüme veri ekleme kısmı beni en zorlayan nokta oldu. “algoritma.cpp” dosyasının 82. Satırında düğüme ilk veriyi ekleme kısmında “head = new Node(item);” kodu beni oldukça zorladı. Tüm uğraşlarımın sonucunda bu satıra “head = new Node(item,NULL,NULL);” parantez içerisine iki adet NULL ekleyerek günlerimi alan hatayı gidermiş oldum. Bu sorunun çözümü için defalarca yeni “Node.hpp” ve “Node.cpp” dosyası oluşturdum. Bir başka zorlandığım kısım ise düğüm içerisindeki verileri main methodu kısmında ascii koda çevirmek oldu. Böyle bir algoritma tasarlamak istedim ve bahsettiğim üzere dizi yapısını da kullanmış oldum.

Bazı bölümler derste yapmış olduğunuz örneklerle oldukça benzemektedir. Sebebi ise ödevimde bilmediğim konularda sizin kaynaklarınız üzerinden ilerlemiş olmamdır. Tüm kodları sıfırdan başlayarak yazdım, hiçbir kopyala yapıştır işlemi olmamıştır.

Programım, yaz okulu başlangıcında vermiş olduğunuz talimatlara uygun olarak kurduğum derleyicide sorunsuz bir şekilde çalışmaktadır. Belirttiğiniz üzere verdiğiniz değerler ile kontrol edilmiştir. Ayrıca algoritmayı kağıt üzerinde kendim çalıştırdım ve sonrasında bu sayıları txt dosyasına ekleyerek programın doğruluğunu tekrar kontrol ettim.

Tüm kontrollerim sonucunda ödevimde eksik bir kısım bulunmamaktadır. Değerlendirmenize sunmaya uygundur.