



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI 1. ÖDEV RAPORU

ÇİFT YÖNLÜ BAĞIL LİSTE KULLANIMI

G151210311 - Ahmet Mücahit GÜRER 2C

SAKARYA

Ağustos, 2020

GİRİŞ

Ödevde Node, Iter, CDLL sınıfları yazılarak başlandı. Node sınıfını yazarken ilk başta **Dugum(const int& veri, Dugum* ileri=NULL, Dugum* geri=NULL)** gibi bir yapı kullanılsa da daha sonra başlık ve kaynak dosyalarını ayırırken sorun yarattı. Parametre göndermeden nesne üretmeye çalıştığımda hata aldım aslında tüm yapıcı metotlar için bu sorunu yaşadım çözüm olarak bazı sınıflarda parametreleri ve parametresiz olarak iki ayrı tanımlama yaparak bu sorunu giderdim.

Aynı şekilde Iter sınıfının kurucu metodunu tanımlarken yaşadım. Orda geliştirdiğim çözüm ise Iter sınıfından nesne türeteceğim zaman

Iter iter;

iter=Iter(basDugum); yerine

Iter iter = Iter(basDugum); -> gibi tanımlayarak ve türeteyeceğim zaman parametre göndererek bu sorunu hallettim. Onun dışında Iter sınıfta takıldığım bir şey olmadı.

CDLL sınıfı, listeyle ilgili tüm işlemlerimi yaptığım algoritmalarımı, hesaplamaları çalıştırdığım bunları gerekli hesaplamalardan sonra listelerime eklediğim ve yazdırdığım sınıf oldu. Açıkcası her işlem(OBEB, Sıralama Algoritması) için sınıf yazıp programı bölmek istemedim. CDLL sınıfında ilk yaptığım bir fonksiyon oluşturup veriyi bunun içinde bir ödevde istenen sıralama algoritmasından geçirip eklemek oldu fakat bunu beceremedim diğer elemanları düzgün eklesede ilk elemanın yerini düzgün eklemiyordu. Bundan vazgeçip Sıralama Algoritması ve Verinin ekleneceği metodları ayırdım. Verinin listedekini yerini ödevdeki gibi istenen şekilde hesaplayacak metodu("**bul**") oluşturdum.

"**bul**" metodunda ilk İteratör'ümü ilgili listenin başına alıyorum ve indexi sıfırlıyorum. Daha sonra girilen değer listenin başındaki değerle OBEB fonksiyonuna parametre olarak veriyorum. Gelen değer o an mevcut olan en büyük obeb değerim olan "**obeb**" değişkeniyle kıyaslıyorum eğer mevcut obebten büyükse bunu yeni obeb olarak kabul ediyorum ve mod işlemlerine geçiyorum. Eğer mevcut obebim gönderdiğim OBEB fonksiyonundan dönen değerden küçükse döngü kuruyorum bu döngüyü indexim liste boyutuna eşit olmayana kadar devam ettiriyorum. Döngü içerisinde bir sonraki elemana geçiyorum ve her seferinde bu şartları kontrol ettiriyorum. Ek olarak hem döngü içerisinde hem döngüye başlamadan hem döngü sonrasında sayılardan birinin 0 olma durumunu kontrol ettiriyorum bu durumlarda sayıları OBEB fonksiyonuna göndermeden değeri 0 kabul ediyorum. Çünkü sayılardan biri 0 olduğunda OBEB ve mod işlemi tanımsız olduğu için program sonlanıyor. Ardından gerekli mod işlemlerini yapıp İteratörümü gerekli konuma getirip **index** diye tanımladığım bir değişken yardımıyla konumu saklıyorum. Aslında "**bul**" metodunda yaptığım kısaca parametre olarak gönderilen verinin listede hangi konuma yerleşeceğini hesaplamak ve bunu **index** değişkeninde saklamak. Daha sonra metodun sonunda hesapladığım konum bilgisini ve veriyi "**ekle**" metoduna gönderiyorum.

"**ekle**" metodunda parametre olarak aldığım konumu, önceki düğümü buldurduğum fonksiyona(**oncekidugumbul**) veriyorum geri dönüş olarak aldığım veriyi **Iter** nesnesinden türettiğim iteratörün içine atıyorum. Daha sonra bu elemanın sonraki ve önceki düğümlerin adresini tutan **ileri**, **geri** isimli pointerları hangi elemanı göstereceklerini ve eklemekte olduğum düğümün hangi düğümü göstereceğini ek olarak gerekiyorsa başlangıç düğümümü ve son düğümümü güncelliyorum. Son olarak ta **size** değişkeninde tuttuğum listemin boyutunu 1 arttırıyorum.

"**obeb**" fonksiyonumda rekürsif bir yaklaşım kullandım açıkcası kod kalabalığını önemli ölçüde azalttığını düşündüğüm için ve derslerimizin konuları arasında da olduğu için ödevde bu şekilde uyguladım. Kullandığım yaklaşımı referanslar kısmına ekledim.

A sayısını B ile böl, kalanı K ile tut,

Eğer kalan K, sıfır ise, cevap B'dir.

Değilse, A yerine B yaz, B yerine de K yaz, ilk adımdan devam et.

SONUÇ

Şablon sınıf yerine int kullanmak o dönüşümleri yapmak biraz uğraştırdı. Çünkü bazı metotlarımda dönüş tipi olarak iki farklı tür olabiliyor. Örnek verecek olursam **ilkveri** ve **sonveri** fonksiyonlarında dönüşün boş olup olmadığı kontrol ettirmem gerekiyor duruma göre int tipinde bir veri yada null döndürmem gerekiyor. Bu sorunu null yerine **hata** adında int bir değişken tanımlayarak ve bunu döndürerek aştım bir hata kontrolü yapmıyorum aslında burada hile yaptım.

Ekrana yazdırma işlemlerinde başlık ve kaynak dosyalarını ayırdıktan sonra sorun yaşadım daha sonra derste hocamız o noktaya değindi. Dersteki gibi başlık dosyasında **friend** etiketi kullanıp kaynak dosyasında direkt ostream diyerek çıktımı aldım. Yalnız burada sorunu çözerken başlık dosyasına **using namespace std** ve **#include<iostream>** eklemek gerekiyor.

Veriyi dosyadan aldıktan sonra boşluklara göre parçalamak için birkaç yöntem mevcut bunlar vektör kullanmak getline kullanmak strtok kullanmak vb. yada kendimiz tüm elemanları kontrol ettirerek buna göre parçalayacak bir fonksiyon yazmak. Vektörün yasak olacağını düşündüğüm için hem de ödevde satır sayısı kadar liste oluşturmam istendiği için satır sayısını da getline ile elde edebileceğimi bildiğim için getline kullandım. Devam edecek olursak **getline(dosya,kelime," ")** şeklinde veriyi parçalayabiliyorsunuz fakat hangi satırın elemanı olduğunu tespit edemiyorsunuz çözüm olarak iç içe iki getline kullandım. İlkinde satırı alıyorum ikincisinde aldığım bu satırı boşluğa göre parçalayıp değişkene aktarıyorum. Yalnız burada da bir sorun çıkıyor aldığınız satırı direkt parametre olarak getline'a veremiyorsunuz çözüm: stringstream. stringstream'le bir nevi elinizde stringi convert ediyorsunuz. İlk olarak satır sayısı kadar liste oluşturmak için önce getline ile satır sayısını aldım, dosyayı okumak için tekrar getline kullandığımda hata aldım çünkü dosya içerisinde index'in konumu dosya sonunda çözüm olarak dosyanın indexini resetlemek için referanslar kısmına da linkini bıraktığım

dosya.clear();

dosya.seekg(0); çözümünü kullandım. Böyle yapmak hem satır sayısı hem de veriyi okuyup parçalamama izin verdi.

Referanslar

- [1] Ders Notları ve Önceki senelerde yapılmış örnekler: <https://ogr.sakarya.edu.tr/tr/SanalSinif/Katil/80386>
- [2] Getline resetleme: <https://stackoverflow.com/questions/5343173/returning-to-beginning-of-file-after-getline>
- [3] Ascii Dönüştürme: <http://www.cplusplus.com/forum/general/74820/>
- [4] OBEB Rekürsyonu: <https://medium.com/@ibrahim.kurce/%C3%B6klid-algoritmas%C4%B1-ile-obe-hesaplama-223a9461293c>
- [5] Sınıf Tasarımı: <https://www.youtube.com/watch?v=xQia4yZABdQ>
- [6] Sınıf dışından Ostream Kullanımı: <https://stackoverflow.com/questions/48290197/ostream-in-class-does-not-have-a-type>
- [7] Dosya işlemleri: <http://www.cplusplus.com/doc/tutorial/files/>