



T.C.
SAKARYA ÜNİVERSİTESİ

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

ÖDEV BAŞLIĞI

G191210025 - Muhammet ÇAĞATAY

SAKARYA
Ağustos, 2021
Veri Yapıları Dersi

Double Linked List

Muhammet aęatay

^a G191210025-1A

Özet

Ödevde problem olarak Sayilar.txt dosyasında # sembolü ile ayrılmış iki tane sayı bulunmaktadır. Bu sayıları ilk olarak üçerli parçalara bölüp bölünen sayıların ilk elemanı 0 ise bunu 1 ile değiştirip çift yönlü bağlantılı listeye atmamız istenmektedir.

Daha sonrasında ise oluşturulan iki listedeki elemanlar sıralı olarak karşılaştırılmalıdır.

Karşılaştırırken Liste1 de ki eleman Liste2 de ki elemandan büyük ise Liste1 e listeyi ters çevirme işlemi uygulanacaktır. Liste1 de ki eleman Liste2 de ki elemana eşit ise Liste2 ye listeyi ters çevirme işlemi uygulanacaktır. Liste1 de ki eleman Liste2 de ki elemandan küçük ise elemanların bulunduğu düğümler yer değiştirecektir. Bu işlemler tamamlandıktan sonra ise Liste1 de ki sayılar ve Liste2 de ki sayılar ekrana yazdırılacaktır.

© 2017 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Double Linked List. , Node , Swap , Reverse

1. GELİŞTİRİLEN YAZILIM

Geliştirilen yazılımda Node ve DoubleLinkedList olmak üzere iki sınıf bulunmaktadır. Node sınıfı özellik olarak item, next ve prev değişkenleri bulundurur. İtem değişkeni düğümün içinde bulunan değeri , next değişkeni bir sonraki düğümün adresini, prev değişkeni ise bir önceki düğümün adresini tutar.

DoubleLinkedList sınıfında ise Node türünde bir head ve size değişkeni bulunmaktadır, Head değişkenin görevi listenin ilk elemanın adresini tutmaktır. Böylelikle tüm düğümlere head değişkeni yardımıyla ulaşılmaktadır. Size değişkeni ise listenin uzunluğunu tutmaktadır. DoubleLinkedList sınıfında bulunan fonksiyonlar ve çalışma prensipleri aşağıda açıklanmıştır.

PrevByPosition: Fonksiyon parametre olarak aldığı index değerinde bulunan düğümün bir önceki düğümün adresini döndürür. Çalışma prensibi ise ilk düğümün başlayarak belirtilen indexe gidene kadar düğümleri gezer ve belirtilen indexe geldiğinde o düğümünden bir önceki düğümü döndürür.

Count: Listenin uzunluğunu döndüren sabit bir fonksiyondur.

IsEmpty: size değişkeninin 0'a eşit olup olmadığını kontrol ederek listenin boş olup olmadığını döndüren fonksiyondur.

Add: add fonksiyonu listenin sonuna eleman eklediği için Insert fonksiyonuna item değişkenini ve listenin size değişkenini gönderir böylelikle listenin sonuna eleman eklenmiş olur.

Insert: Parametre olarak aldığı itemi belirtilen indexe yerleştirir. Çalışma prensibi ise index değeri sıfıra eşit ise yeni düğümü direk head e atar ve next değişkenine de önceki headin adresini atar böylelikle listenin başına eklenmiş olur. Index değeri sıfıra eşit değilse PrevByPosition fonksiyonu yardımıyla belirtilen indexten bir önceki düğümü bulur ve next'ine yeni düğümü yerleştirir.

First: Listenin ilk elemanını döndürür.

Last: Listenin son elemanını döndürür.

IndexOf: Parametre olarak aldığı itemin indexini döndürür.

Remove: Parametre olarak aldığı itemin bulunduğu düğümü siler.

RemoveAt: Parametre olarak aldığı indexde bulunan düğümü siler.

Find: Parametre olarak aldığı itemin listede bulunup bulunmadığını kontrol eder.

ElementAt: Parametre olarak aldığı indexde bulunan itemi döndürür.

Reversing: Listeyi ters çeviren fonksiyondur. Çalışma prensibi ise ilk olarak son düğümü bulur ve bunu listenin head değişkenine atar daha sonra ise while döngüsü ile ilk elemana gelene kadar tüm düğümlerin next ve prev değerlerini değiştirir. Böylelikle liste ters çevirilmiş olur.

Clear: Listeyi temizleme fonksiyonudur.

Swap: İki liste arasındaki düğümleri değiştiren fonksiyondur. Fonksiyonun çalışma prensibi parametre olarak diğer listenin değişecek düğümü, kendi listesinin değişecek düğümünün next ve prev değişkenlerini ve değişecek düğümlerin index değerini alır. Daha sonra index değeri sıfıra eşit ise head değişkenine diğer listenin düğümünü atar ve bu düğümün next'i ne önceki head'in next'i ni , next değeri null değil ise next'in previne de head değişkenini atar böylelikle çift yönlü bağlanmış olur. Index değeri sıfıra eşit değil ise değişecek düğümün next i ne listede bulunan next , prev'i ne ise listede bulunan prev atanır. Next değişkeni null değil ise next'in previne değişecek düğüm atanır. Aynı şekilde prev değişkeni null değil ise prev'in next'i ne değişecek olan düğüm atanır. Böylelikle listeler arasında düğüm değiştirme işlemi uygulanmış olur.

Test.cpp dosyasında 3 adet fonksiyon bulunmaktadır. Bunlardan ilki read file fonksiyonudur. Fonksiyon parametre olarak aldığı dosya ismini alır ve bu dosyayı okuyup bir string değişkenine atar.

İkinci fonksiyon ise groupBy fonksiyonudur. Bu fonksiyon okunan dosyadaki stringi alır ve üçerli gruplara parçalar bu üçerli gruplardaki sayıların yüzler basamağı 0 ise 1 ile değiştirir ve DoubleLinkedList'e atar ve oluşan listeyi döndürür.

Üçüncü fonksiyon ise compare fonksiyonudur. Bu fonksiyon parametre olarak 2 tane DoubleLinkedList alır ve bu listeleri ilk indexden son indexe kadar karşılaştırır. Eğer liste1 de ki düğümün değeri liste2 de ki düğümün değerinden büyük ise liste1'in ters çevirme fonksiyonu çağrılır. Liste1 de ki ile liste2 de ki eşit ise liste2 nin ters çevirme fonksiyonu çağrılır. Liste1 de ki liste2 de ki değerden küçük ise düğümler yer değiştirilir.

2. ÇIKTILAR

Girdi: 122654893219849179258147058126#741158654078987159357258369173

Çıktı:

```
Sayı1: 741654893357987179258654369173  
Sayı2: 122158147178849159219258158126  
PS C:\Users\MuhammetÇağatay\Desktop\veriyapıları\odev>
```

Girdi: 058#012

Çıktı

```
Sayı1: 158  
Sayı2: 112  
PS C:\Users\MuhammetÇağatay\Desktop\veriyapıları\odev>
```

Girdi: 815102162524#622101830754

Çıktı:

```
Sayı1: 815102830754  
Sayı2: 622101162524  
PS C:\Users\MuhammetÇağatay\Desktop\veriyapıları\odev>
```