

## VERİ YAPILARI 2020-2021 GÜZ DÖNEMİ 1. ÖDEV RAPORU

### İstenilenler:

Ödevde öncelikle istenilen program açıldığı gibi “Sayılar.txt” dosyasının okunup, her satır için sayı adedinin tek olduğunun, her satırda en az üç adet sayı olduğunun ve 0’dan küçük sayı bulunmadığının kontrolünün yapılarak bu şartların sağlanmadığı durumda programın sona erdirilmesidir. Şartlar program açıldığında istenildiği gibi her satır için ayrı İki Yönlü Dairesel Bağlı Liste oluşturulması ve bu listelerde liste başı veya sonu gibi göstericilerin bulunmaması, yalnızca liste ortası göstericisinin bulunması istenmektedir. Aynı zamanda oluşturulan her satır için ayrı bağlı listenin de bir yapının içerisinde tutulması gerekmektedir. Bu istenilen liste orta göstericisi her bağlı liste için metin belgesinden okunan ilk sayı olacaktır. Liste ortasının gerisine satırdaki sayıların ortasına kadar olan sayılar sırasıyla, ilerisine ise ortasından sonraki sayılar sırasıyla düğüm olarak eklenecektir. Satır sayısı adedince bağlı liste bağlı listenin ekle metoduna dizi olarak gönderilerek oluşturulduktan sonra bu listelerden liste ortası düğümünün değeri en büyük ve en küçük olan liste seçilecek (minimum veya maksimum sayıda eşitlik olması durumunda herhangi biri alınabilir) ve büyük listeOrta’nın bulunduğu listenin solundaki değerlerin küçük listeOrta’nın sağındaki düğümler ile, sağındaki değerlerin küçük listeOrta’nın solundaki düğümler ile ters sıralı şekilde yer değiştirilerek çaprazlama yapılması istenmektedir. Program bu işlemlerin en sonunda ise konsola çıktı olarak bu iki listenin son hallerini ve bu listelerin listeOrta göstericilerinin heap bellek bölgesindeki adreslerini verecektir.

### Yaptıklarım:

Öncelikle Node ve CircularDoublyLinkedList isimli sınıfları ve bu sınıflara ait gerekecek fonksiyonları ve metodları oluşturdum. Listelerdeki düğümler için kullanacağım Node sınıfında düğümün öncesini ve sonrasını gösteren pointerları, datasını ve kurucu metodunu oluşturdum. Ana main dosyasında ödevde istenildiği şekilde programın açıldığı anda metin dosyasını okutup öncelikle satırlarda gerekli kontrolleri sağlayarak istenilmeyen durumlarda programın sona erdirilmesi sağladım. Şartlar sağlanıyorsa oluşturulacak liste adedini belirlemek için döngü ile satır sayısını bulup,

```
CircularDoublyLinkedList **listeler = new CircularDoublyLinkedList *[satirSayisi];
```

şeklinde listelerin tutulacağı yapıyı oluşturdum. Hemen sonrasında ise yine satırlarda dolaşarak öncelikle satırdaki sayı sayısını bulup her satır için geçici dinamik bir dizi oluşturdum ve CircularDoublyLinkedList sınıfında add metodunu geliştirerek satırdaki sayıları alıp metodun içinde öncelikle dizinin 0. indeksini listeOrta göstericisi olarak tanımlayarak sonrasında dizinin indekslerinde dönerek gerisine ve ilerisine CircularDoublyLinkedList sınıfında oluşturduğum FindNextNode ve FindPreviousNode fonksiyonlarını kullanarak istenilen şekilde düğümlerini oluşturdum ve ekledikten sonra aynı döngüde bulunan satır için oluşturulan bu geçici dinamik listeyi heap bölgeden sildim. Bir önceki cümlede bahsettiğim tek bir liste için olayı her satır için ayrı liste için bir sayaç ile her satırda 1 ekleyerek listelerin 0. indeksinden itibaren satır sayısı kadar listeyi şu şekilde metin belgesinde her satırı dolaşırken oluşturdum.

```
listeler[sayac]=new CircularDoublyLinkedList();
```

```
listeler[sayac]->add(satirdakiSayilarDizisi,satirdakiSayiSayisi);
```

Satır sayısı kadar ayrı bağlı liste oluşturulduktan sonra ise CircularDoublyLinkedList sınıfında oluşturduğum listelerden minimum listeOrta’yı ve maksimum listeOrta’nın bulunduğu listenin indeksini veren minIndexBul ve maxIndexBul fonksiyonlarını kullanarak bulunan değeri alıp main dosyada yazımı kolay olması açısından minIndex ve maxIndex değişkenlerine atadım.

Çaprazlama işlemi için ise çaprazlandığında düğümlerin diğer bir listeye ters şekilde sıralanması gerektiğinden öncelikle CircularDoublyLinkedList sınıfında reverseOrder isimli bir metot oluşturarak çaprazlamadan sonra geleceği sıra şeklinde listenin ortasından gerisini ve ilerisini yeniden sıraladım. CircularDoublyLinkedList sınıfında son işlemi gerçekleştirecek oluşturduğum, öncelikle maksimum listeOrta ve minimum listeOrta'nın bulunduğu listeleri reverseOrder metoduyla ters çevirerek sonraki işlemler için daha kolay hale getirerek, gerekli kodları yazarak düğüm yerlerini gerekli listeden gerekli yerlerden koparıp diğer bir listenin gerekli yerlerine ve diğer düğüm bağlantılarını da sağlayarak işlemi main dosyasında da çaprazlama metodunu çağırarak gerçekleştirdim. Bu işlemin sonunda da minimum listeOrta'nın bulunduğu liste ile maksimum listeOrta'nın bulunduğu listelerin sizelerini değiştirerek çaprazlama işlemi tamamlamış oldum.

Programa çıktı verirken gereken listeOrta değerlerinin heap bellek bölgesindeki adresini görebilmek için CircularDoublyLinkedList sınıfında listeOrtaAdres fonksiyonunu oluşturdum ve dokümanda istenildiği şekilde en küçük, en büyük listenin listeOrta'larının heap bellek bölgesindeki adreslerini ve listelerin son hallerini ekrana yazdırdım.

Son olarak ise oluşturulan listelere for döngüsüyle liste adedince indeks indeks erişerek yıkıcı metotlarında bulunan CircularDoublyLinkedList sınıfındaki clear metoduyla tüm düğümlerini silip aynı zamanda listelerin listeOrta'larını da heap bölgesinden sildim. Sonrasında ise listelerin tutulduğu yapı olan listeler'i de heap bölgesinden silerek ödevi bitirdim.

### **Öğrendiklerim:**

Derste öğrenmiş olduğum liste yapısını fonksiyonları geliştirerek ve yeni fonksiyonlar ekleyerek, çaprazlama esnasında daha da pekiştirerek bu yapıda hemen hemen her şeyi yapabilecek düzeye geldim.

### **Zorlandığım kısımlar:**

Ödev başlama konusunda çok zorlandım, kafamda yapıyı oluşturmak fikir aşaması açısından biraz uzun sürdü. Aslında çok basit bir şey olduğunu sonradan araştırıp bulsam da listelerin başka bir yapıda tutulmasını halledene kadar zorlandım fakat daha sonrasında başladığımda ders gruplarında en zorlanıldığı söylenilen çaprazlama konusunu da listelerin çaprazlama öncesi ters çevirmeyi akıl ettiğimde kolayca aşabildim.

### **Eksik bıraktığım yerler:**

Her cümleyi dikkatle okuyarak ödevi yaptım, bireysel olarak beceremediğim, yapamadığım veya cümlelerde kaçırduğım bir yer olmadı. Eksik bıraktığım bir yer bulunmamaktadır.