



Yıldız Teknik Üniversitesi
Elektrik Elektronik Fakültesi
Bilgisayar Mühendisliği

BLM2512
Veri Yapıları Ve
Algoritmalar
GR:2

DR.ÖĞRETİM ÜYESİ GÖKSEL BİRİCİK

ÖDEV-3

İsim: Muhammed Taha Güneş
No: 21011017

Video Linki:

https://www.youtube.com/watch?v=ZHWQAm_STDM

E.posta: taha.gunes@std.yildiz.edu.tr

Problemin Tanımı

Matrisin her bir satırında bir kuyruk olacak şekilde birden fazla kuyruktan önceliği yüksek olanlar sırasıyla bulunması amaçlanmıştır.

Problemin Çözümü (Algoritma)

Gerekli tanımlamalar ve fonksiyonlar yazılmıştır. İlk olarak kullanıcıdan matrisin satır ve sütun sayısı alınmıştır. Daha sonra kullanıcıdan satır sayısı kadar yani her bir kuyruk için kapasite miktarları alınmıştır. Eğer M yani sütun sayısından fazla kapasite girilmişse kullanıcıdan tekrardan kapasite girilmesi istenilmiştir. Matrisi 1'den NXM sayısına kadar rastgele doldurma işlemi ayrı bir fonksiyonda yapılmıştır. Bu fonksiyonda farklı sayılar isminde bir dizi oluşturulmuştur. 1'den NXM sayısına kadar sırasıyla değerler atanmıştır. Rand fonksiyonu ile bu değerler arasında rastgele bir sayı oluşturulup bu sayı rastgele indis değişkenine atanmıştır. O indisteki sayı ile yeri değiştirilmiştir. Bu şekilde tüm dizi rastgele bir şekilde karıştırılmıştır. Karıştırma işleminden sonra bu değerler kapasite miktarı kadar matrisimize atanmıştır. Kapasiteden fazla olan yerler ise boşluğu gösteren -1 değerleri ile doldurulmuştur. Oluşan matrisin tüm satırları yani tüm kuyruklara maxHeapify işlemi uygulanmıştır. Sonrasında ise öncelik sıralama fonksiyonumuz çalıştırılmıştır. Bu fonksiyon asıl işlemlerimizin yapıldığı fonksiyondur. Bitme kontrol fonksiyonu 1 olana kadar döngü oluşturulmuştur. Bitme kontrol fonksiyonu ise tüm kuyrukların boyutları 0 ise 1 döndürmektedir. Diğer şartlarda döngü devam ettirilmiştir. Döngü içerisinde max eleman bulma fonksiyonu çağırılmıştır. Bu fonksiyon tüm kuyrukların ilk sıradaki yani başındaki elemanlar arasından en büyük olanı bulup onun satır yani kuyruk indisini döndürmüştür. Eğer kuyruğun başı boşsa yani -1 ve boyut matrisinde o indisteki değer 0 ise o satırı atlamıştır. Fonksiyonun sonucunda max elemanın olduğu indis sonraki fonksiyonumuz olan max eleman silme fonksiyonu için lazım olmuştur. Max eleman silme fonksiyonunda ise o kuyruk boş değilse (boyut dizisi 0 değilse) tüm elemanları bir birim sola ötelenip en sağa boşluk anlamına gelen -1 eklenmiş ve boyut dizisindeki o indisteki değer 1 azaltılmıştır. Hemen sonrasında boyut dizisinin o değeri 0 ise yani kuyruk boşsa o kuyruğun indisi bitiş dizisine atılmıştır. Eğer kuyruk doluysa o kuyruk sonraki karşılaştırmalar için tekrardan maxHeapify edilmiştir. Her adımda matris yazdırılmıştır. Tüm kuyruklar boşsa döngü bitirilmiştir. En sonda ise bitiş dizisi bitme sırasını indis tipinde tuttuğu için onu sıraya çevirmek için 1 ile toplanarak sırasıyla ekrana basılmıştır.

Karşılaşılan Sorunlar

İlk karşılaşılan sorun maxHeapTree fonksiyonunda maxHeapify yaparken parametre olarak ne verilmesi gerektiği olmuştur. Bu sorun ilk parametre olarak matrisin o satırı yani işlem yapılan kuyruk, diğer parametre olarak o kuyruğun kapasitesini tutan dizi elemanının verilmesi ile çözülmüştür. 2.sorun kuyruk öncelikleri sıralanırken while döngüsünün sonsuz döngüye girmesi olmuştur. İlk düşünülen fikir, boyut dizisinde herhangi bir değişiklik yapmadan -1 yani boşluklar üzerinden bitme kontrolü yapılmasıdır. Bu fikrin uygulaması başarılı olamamıştır. Bunun yerine bitme kontrolü boyut dizisinin o indisi 0 yani o kuyruk bitmişse şeklinde yapılmıştır.

Ekran Görüntüleri

Senaryo 1: (M değerinden büyük kapasite girilirse)

```
Matris satir sayisini giriniz.  
2  
Matris sutun sayisini giriniz.  
2  
Her bir oncelikli kuyrugun kapasitesini girin (En fazla M degeri kadar girebilirsiniz):  
2  
5  
Hata: Kapasite M degerinden buyuk olamaz. Lutfen tekrar girin:
```

Senaryo 2:

```
Matris satir sayisini giriniz.  
4  
Matris sutun sayisini giriniz.  
5  
Her bir oncelikli kuyrugun kapasitesini girin (En fazla M degeri kadar girebilirsiniz):  
2  
3  
4  
4  
  
19 12 -1 -1 -1  
10 4 16 -1 -1  
5 9 3 11 -1  
2 1 6 7 -1  
  
Tum kuyruklar heapify edildikten sonraki hali=  
  
19 12 -1 -1 -1  
16 4 10 -1 -1  
11 9 3 5 -1  
7 2 6 1 -1  
  
Kuyruklardan degeri okunan elemanlar=19 16 11 7  
Secilen eleman = 19  
  
12 -1 -1 -1 -1  
16 4 10 -1 -1  
11 9 3 5 -1  
7 2 6 1 -1  
  
Kuyruklardan degeri okunan elemanlar=12 16 11 7  
Secilen eleman = 16  
  
12 -1 -1 -1 -1  
10 4 -1 -1 -1  
11 9 3 5 -1  
7 2 6 1 -1  
  
Kuyruklardan degeri okunan elemanlar=12 10 11 7  
Secilen eleman = 12  
  
-1 -1 -1 -1 -1  
10 4 -1 -1 -1  
11 9 3 5 -1  
7 2 6 1 -1  
  
Kuyruklardan degeri okunan elemanlar=10 11 7  
Secilen eleman = 11  
  
-1 -1 -1 -1 -1  
10 4 -1 -1 -1  
9 3 5 -1 -1  
7 2 6 1 -1  
  
Kuyruklardan degeri okunan elemanlar=10 9 7  
Secilen eleman = 10
```

```
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
9 3 5 -1 -1
7 2 6 1 -1
```

Kuyruklardan degeri okunan elemanlar=4 9 7
Secilen eleman = 9

```
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
5 3 -1 -1 -1
7 2 6 1 -1
```

Kuyruklardan degeri okunan elemanlar=4 5 7
Secilen eleman = 7

```
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
5 3 -1 -1 -1
6 2 1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=4 5 6
Secilen eleman = 6

```
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
5 3 -1 -1 -1
2 1 -1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=4 5 2
Secilen eleman = 5

```
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
3 -1 -1 -1 -1
2 1 -1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=4 3 2
Secilen eleman = 4

```
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
3 -1 -1 -1 -1
2 1 -1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=3 2
Secilen eleman = 3

```
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
2 1 -1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=2
Secilen eleman = 2

```
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
1 -1 -1 -1 -1
```

Kuyruklardan degeri okunan elemanlar=1
Secilen eleman = 1

```
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
```

Kuyrukların bitis sirasi=
1 2 3 4

Senaryo 3:

```
Matris satir sayisini giriniz.
2
Matris sutun sayisini giriniz.
5
Her bir oncelikli kuyrugun kapasitesini girin (En fazla M degeri kadar girebilirsiniz):
3
4

4 3 8 -1 -1
1 10 7 2 -1

Tum kuyruklar heapify edildikten sonraki hali=

8 3 4 -1 -1
10 2 7 1 -1

Kuyruklardan degeri okunan elemanlar=8 10
Secilen eleman = 10

8 3 4 -1 -1
7 2 1 -1 -1

Kuyruklardan degeri okunan elemanlar=8 7
Secilen eleman = 8

4 3 -1 -1 -1
7 2 1 -1 -1

Kuyruklardan degeri okunan elemanlar=4 7
Secilen eleman = 7

4 3 -1 -1 -1
2 1 -1 -1 -1

Kuyruklardan degeri okunan elemanlar=4 2
Secilen eleman = 4

3 -1 -1 -1 -1
2 1 -1 -1 -1

Kuyruklardan degeri okunan elemanlar=3 2
Secilen eleman = 3

-1 -1 -1 -1 -1
2 1 -1 -1 -1

Kuyruklardan degeri okunan elemanlar=2
Secilen eleman = 2

-1 -1 -1 -1 -1
1 -1 -1 -1 -1

Kuyruklardan degeri okunan elemanlar=1
Secilen eleman = 1

-1 -1 -1 -1 -1
-1 -1 -1 -1 -1

Kuyrukların bitis sirasi=
1 2
-----
Process exited after 4.709 seconds with return value 0
Press any key to continue . . . _
```

Video Link:

https://www.youtube.com/watch?v=ZHWQAm_STDM