

1- Problem Tanımı:

$N \times M$ 'lik bir matrisin her satırında max-heap-tree özelliği taşıyan öncelikli kuyruk oluşturulması istenmektedir. Öte yandan her bir kuyruğun kapasitesi farklıdır. Matrisin sütun sayısı ise öncelikli kuyruklardan en yüksek kapasiteye sahip olana göre belirlenmelidir. Her adımda öncelikli kuyrukların ilk elemanlarından en yüksek önceliğe sahip olan ilgili kuyruktan çıkartılır. Bu adımlar kuyruklarda eleman kalmayınca kadar devam eder. Tasarlayacağınız algoritma sıra ile boşa çıkan kuyrukları ekrana yazdırmalıdır

2- Problemin Çözümü:

Kullanıcının girmiş olduğu satır ve sütun sayısı ile oluşturulan bir matris, yine kullanıcıdan alınan kuyruk boyutlarına göre rastgele üretilen çoklu öncelikli kuyruk yapılarıyla doldurulmuştur. Daha sonrasında bu kuyruk yapılarına max heap tree özelliği kazandırılmıştır. Düzenlenen matris ekrana yazdırıldıktan sonra ilk sütundaki en büyük elemanı içeren kuyruk bulunmuş ve en büyük eleman silinmiştir. Daha sonrasında eleman silinen kuyruk boş değilse tekrardan max heap tree özelliğine göre sıralandırılmıştır, kuyruğun boş olması durumunda ise kuyruğun sırası bir dizide tutulmuştur, tüm kuyruklar boşalana kadar bu adımlar tekrarlanmış, en sonda ise kuyrukların hangi sırayla boşaltıldığı ekrana yazılmıştır.

3- Karşılaşılan Sorunlar:

Matrisi rastgele elemanlarla doldururken, aynı sayıların kullanılması durumuna karşı rastgele üretilen sayının daha önce matriste olup olmadığı kontrol edilmiştir.

Matris doldurulurken kuyruk yapısının uzunluğunun matris sütunundan küçük olması durumunda boş kalan yerler -1 değeriyle doldurulmuştur. Kuyruk boyutlarının sütun sayısından büyük olmaması için de kullanıcıya uyarı verilmiştir.

Eleman çıkartılan kuyrukların boş olup olmadıkları kontrol edilmiştir ve boş olan kuyruklar dizide tutulmuştur.

Matrisin ilk sütunundaki eleman -1 ise karşılaştırılma işlemine dahil edilmemiştir.

4- Ekran Çıktıları:

Senaryo 1-

```
Matrisin satir sayisi(M) girin: 2
Matrisin sutun sayisini(M) girin: 3
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 3'den (sutun sayisindan)buyuk olmamali
3
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 3'den (sutun sayisindan)buyuk olmamali
3
Rastgele uretilen 2 kuyruk iceren matris
3 4 1
6 5 2
Max heap ozelligi tasiyan 2 kuyruk iceren matris
4 3 1
6 5 2
Kuyruklardan degerleri okunan elemanlar:4 6
Secilen eleman 6
4 3 1
5 -1 2
Kuyruklardan degerleri okunan elemanlar:4 5
Secilen eleman 5
4 3 1
2 -1 -1
Kuyruklardan degerleri okunan elemanlar:4 2
Secilen eleman 4
3 -1 1
2 -1 -1
Kuyruklardan degerleri okunan elemanlar:3 2
Secilen eleman 3
1 -1 -1
2 -1 -1
Kuyruklardan degerleri okunan elemanlar:1 2
Secilen eleman 2
1 -1 -1
-1 -1 -1
Kuyruklardan degerleri okunan elemanlar:1
Secilen eleman 1
-1 -1 -1
-1 -1 -1
1
0
-----
Process exited after 5.341 seconds with return value 0
Press any key to continue . . .
```

Senaryo 2-

```
Matrisin satir sayisi(M) girin: 4
Matrisin sutun sayisini(M) girin: 5
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 5'den (sutun sayisindan)buyuk olmamali
5
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 5'den (sutun sayisindan)buyuk olmamali
5
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 5'den (sutun sayisindan)buyuk olmamali
5
Oncelikli kuyruk boyutlarini girin.Kuyruk boyutu 5'den (sutun sayisindan)buyuk olmamali
5
Rastgele uretilen 4 kuyruk iceren matris
11 16 9 -1 -1
4 20 8 -1 -1
19 2 5 17 -1
12 10 1 7 14
Max heap ozelligi tasiyan 4 kuyruk iceren matris
16 11 9 -1 -1
10 4 8 -1 -1
19 17 5 2 -1
14 12 1 7 10
Kuyruklardan degerleri okunan elemanlar:16 20 19 14
Secilen eleman 20
16 11 9 -1 -1
8 4 -1 -1 -1
19 17 5 2 -1
14 12 1 7 10
Kuyruklardan degerleri okunan elemanlar:16 8 19 14
Secilen eleman 19
16 11 9 -1 -1
8 4 -1 -1 -1
17 2 5 -1 -1
14 12 1 7 10
Kuyruklardan degerleri okunan elemanlar:16 8 17 14
Secilen eleman 17
16 11 9 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
14 12 1 7 10
Kuyruklardan degerleri okunan elemanlar:16 8 5 14
Secilen eleman 16
11 -1 9 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
14 12 1 7 10
Kuyruklardan degerleri okunan elemanlar:11 8 5 14
Secilen eleman 14
11 -1 9 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
12 10 1 7 -1
```

```

Kuyruklardan degerleri okunan elemanlar:11 8 5 12
Secilen eleman 12
11 -1 -1 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
10 7 1 -1 -1
Kuyruklardan degerleri okunan elemanlar:11 8 5 10
Secilen eleman 11
9 -1 -1 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
10 7 1 -1 -1
Kuyruklardan degerleri okunan elemanlar:9 8 5 10
Secilen eleman 10
9 -1 -1 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
7 -1 1 -1 -1
Kuyruklardan degerleri okunan elemanlar:9 8 5 7
Secilen eleman 9
-1 -1 -1 -1 -1
8 4 -1 -1 -1
5 2 -1 -1 -1
7 -1 1 -1 -1
Kuyruklardan degerleri okunan elemanlar:8 5 7
Secilen eleman 8
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
5 2 -1 -1 -1
7 -1 1 -1 -1
Kuyruklardan degerleri okunan elemanlar:4 5 7
Secilen eleman 7
-1 -1 -1 -1 -1
4 -1 -1 -1 -1
5 2 -1 -1 -1
1 1 -1 -1 -1
Kuyruklardan degerleri okunan elemanlar:4 5 1
Secilen eleman 5
-1 -1 -1 -1 -1
6 -1 -1 -1 -1
2 -1 -1 -1 -1
1 -1 -1 -1 -1
Kuyruklardan degerleri okunan elemanlar:4 2 1
Secilen eleman 4
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
2 -1 -1 -1 -1
1 -1 -1 -1 -1

```

```

Kuyruklardan degerleri okunan elemanlar:2 1
Secilen eleman 2
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
-1 -1 -1 -1 -1
1 -1 -1 -1 -1
Kuyruklardan degerleri 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
0
1
2
3
-----
Process exited after 12.67 seconds with return value 0
Press any key to continue . . .

```