



BLM2512 Veri Yapıları ve Algoritmalar

**Çoklu Öncelikli Kuyruk Gerçeklenmesi**

**Öğrenci Adı Soyadı :** *Berat Can KARAKAŞ*

**Öğrenci No :** *21011102*

**Dersin Eğitmeni :** *Mine Elif KARSLIGİL*

**Video Linki :** <https://www.youtube.com/watch?v=Efzr1hzbX6A>

# 1 - Problem Tanımı ---

Bu problemde kullanıcıdan alınan satır ve sütun sayıları ile bir matris oluşturulur daha sonra kullanıcıdan her bir satırın kapasitesi istenmektedir . Ondan sonra ise matristeki kapasiteleri kadar alanı dolduracak şekilde eşsiz sayılar rastgele şekilde matrise yerleştirilir . Bundan sonra yapılacak her işlemde sonra matrisin her bir satırı kendi içinde max heap tree özelliği taşıyacak şekilde güncellemeler yapılır . Her satır max heap olduktan sonra her satırdaki en büyük eleman seçilir ve aralarından en büyük eleman çıkarılır . Çıkarılan satır max heap özelliği taşımadığı için heapify işlemi yapılır ve tekrar heap özelliği kazandırılır . Bu işlem matriste çıkarılacak eleman kalmayana kadar devam eder . ( Matristen çıkarılan elemanların yerine boş olduğunu belirtmek için -1 koyulmaktadır . )

# 2 - Problem Çözümü ---

Bu problemde problemde kullanıcıdan matrisin satır , sütun sayısı ve her satırın kapasitesi istenmiştir . Bu işlemde sonra her satırın kapasitesi kadar eleman aynı sayı tekrar matrise yerleştirilemeyecek şekilde rastgele seçilip matrise yerleştirilir . Bu adımdan sonra her bir satırı kendi içinde max heap yapan bir fonksiyon ile başlangıçtaki matrisin tamamı heapify edilir . Bu fonksiyon her satırın 0. indisinden başlar , çocukları arasından büyük olanı bulur ve kendisi ile kıyaslar . Eğer kendisinden büyük ise çocuğu , çocuğu ile kendisinin yerini swap fonksiyonu ile değiştirir ve indisi bir üst sıraya yani kendi parent ı varsa oraya eşitler ( yeni indis = (eski indis - 1)/2 . Her değişme işleminden sonra bu şekilde devam edilir . değişme işlemi olmadığı taktirde indis 1 arttırılır .

Bu işleme kapasiteye erişilinceye kadar devam edilir .  
Fonksiyondan çıkıldıktan sonra her satır kendi içinde max heap özelliği kazanmış olacaktır . Daha sonra Her satırın 0. indisi seçilir ve arasından en büyük eleman bulunup matrizen çıkarılır ve yerine -1 koyulur . Silme işlemi yapılan satır tekrar heapify fonksiyonuna gönderilir ve tekrar heap özellik kazandırılır . Matriste çıkarılacak eleman kalmayana , tüm elemanlar -1 olana kadar bu işleme devam edilir ve her satırın bitiş sırası bir dizide tutulur , işlemler tamamen bittikten sonra bitiş sırası ekrana yazdırılır ve fonksiyon tamamlanır .

### 3 - Örnek Çıktılar ---

N = 5 , M = 4 , Kapasite = 1,2,3,4,4

Çıktı = Bitiş Sırası : 5,1,4,3,2 . satırlar

```
-- Main Array --
8      -1      -1      -1
16     1       -1      -1
10     12      3       -1
9       7       17      13
11     19      20      18

--

-- First Heapify --

8      -1      -1      -1
16     1       -1      -1
10     12      3       -1
9       7       17      13
11     19      20      18

8      -1      -1      -1
16     1       -1      -1
10     12      3       -1
9       7       17      13
11     19      20      18

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
9       7       17      13
11     19      20      18

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
11     19      20      18

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
20     19      11      18
```

```
--
8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
20     19      11      18

20 - removed

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
19     11      18      -1

--

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
19     11      18      -1

19 - removed

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
18     11      -1      -1

--
```

```
--
8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
18     11      -1      -1

18 - removed

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
11     -1      -1      -1

--

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
17     13      9       7
11     -1      -1      -1

17 - removed

8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
13     9       7       -1
11     -1      -1      -1

--
```

```

--
8      -1      -1      -1
16     1       -1      -1
12     10      3       -1
13     9       7       -1
11     -1      -1      -1

16 - removed

8      -1      -1      -1
1      -1      -1      -1
12     10      3       -1
13     9       7       -1
11     -1      -1      -1

--

--

8      -1      -1      -1
1      -1      -1      -1
12     10      3       -1
13     9       7       -1
11     -1      -1      -1

13 - removed

8      -1      -1      -1
1      -1      -1      -1
12     10      3       -1
9      7       -1      -1
11     -1      -1      -1

--

```

```

--
8      -1      -1      -1
1      -1      -1      -1
12     10      3       -1
9      7       -1      -1
11     -1      -1      -1

12 - removed

8      -1      -1      -1
1      -1      -1      -1
10     3       -1      -1
9      7       -1      -1
11     -1      -1      -1

--

--

8      -1      -1      -1
1      -1      -1      -1
10     3       -1      -1
9      7       -1      -1
11     -1      -1      -1

11 - removed

8      -1      -1      -1
1      -1      -1      -1
10     3       -1      -1
9      7       -1      -1
-1     -1      -1      -1

--

```

```

--
8      -1      -1      -1
1      -1      -1      -1
10     3       -1      -1
9      7       -1      -1
-1     -1      -1      -1

10 - removed

8      -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
9      7       -1      -1
-1     -1      -1      -1

--

--

8      -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
9      7       -1      -1
-1     -1      -1      -1

9 - removed

8      -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
7      -1      -1      -1
-1     -1      -1      -1

--

```

```

--
8      -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
7      -1      -1      -1
-1     -1      -1      -1

8 - removed

-1     -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
7      -1      -1      -1
-1     -1      -1      -1

--

--

-1     -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
7      -1      -1      -1
-1     -1      -1      -1

7 - removed

-1     -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
-1     -1      -1      -1
-1     -1      -1      -1

--

```

```

--
-1     -1      -1      -1
1      -1      -1      -1
3      -1      -1      -1
-1     -1      -1      -1
-1     -1      -1      -1

3 - removed

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

--

--

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

1 - removed

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

--

Finishing Order --> 5 , 1 , 4 , 3 , 2 ,

```

Sarı olan  
kısımlarda  
seçme yeşil  
olan  
kısımlarda  
heapify  
işlemi  
yapılmakta  
dır .

N = 3 , M = 7 , Kapasite = 0,2,7

Çıktı = Bitiş Sırası : 1,2,3 . satırlar

```
-- Main Array --  
-1      -1      -1      -1      -1      -1      -1  
6       11      -1      -1      -1      -1      -1  
3       17      14      20      10      18      8  
  
--  
  
-- First Heapify --  
-1      -1      -1      -1      -1      -1      -1  
6       11      -1      -1      -1      -1      -1  
3       17      14      20      10      18      8  
  
-1      -1      -1      -1      -1      -1      -1  
11      6       -1      -1      -1      -1      -1  
3       17      14      20      10      18      8  
  
-1      -1      -1      -1      -1      -1      -1  
11      6       -1      -1      -1      -1      -1  
20      17      18      3       10      14      8  
  
--
```

```
--  
-1      -1      -1      -1      -1      -1      -1  
-1      -1      -1      -1      -1      -1      -1  
3       -1      -1      -1      -1      -1      -1  
  
3 - removed  
  
-1      -1      -1      -1      -1      -1      -1  
-1      -1      -1      -1      -1      -1      -1  
-1      -1      -1      -1      -1      -1      -1  
  
--  
  
Finishing Order --> 1 ,2 ,3 ,
```