

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Dersin Adı**

2017 2018 Bahar Yılı  
Veri Yapıları ve Algoritmalar

**Ödev-3**

**Ödevin Konusu**

Graflar

**Dersi Veren Öğretim Üyesi**

Doç. Dr. M.Elif KARSLIGİL

**Ödevi Yapan Öğrenci**

Emre Çelik

15011902

**Problem:** Nesnelerin birbirlerine benzerliklerine göre gruplanması(clustering) Makine Öğrenmesinin önemli konularından birisidir. Dokümanların içerik benzerliklerine göre gruplanması, tıbbi görüntülerin benzerliklerine göre gruplanması bu konuda yapılan uygulamalara örnek verilebilir.

Bu ödevde birbirlerine benzerlik mesafesi verilen farklı nesneleri **Minimum Spanning Tree** yöntemini kullanarak birbirine en çok benzeyen nesnelerin bir arada bulunduğu **k adet kümeye yerleştiren** bir uygulama tasarlanıp gerçekleştirilecektir.

#### İşlem Adımları:

1. Sırası ile en yakın mesafeli iki nesneyi MST algoritması ile aynı kümeye alınız.
2. K adet küme elde edene kadar işleme devam ediniz.
3. Her kümedeki nesneleri(yani düğümleri) yazdırınız.
4. Verilen iki küme arasındaki en kısa mesafeyi(İki kümedeki en yakın düğümler arasındaki mesafedir) bulunuz.

**Veri Dosyası:** data.txt dosyasından veriler çekilecektir.

```
data.txt
1,0,4,0,0,2,4,0,0,1,2,0,2,8,8,5,0,3,7,8,0,0,10,9,6,5,1,1,5,0
0,1,6,6,9,8,0,4,3,0,9,4,6,0,10,0,3,0,0,10,8,7,4,4,0,0,0,10,0,0
4,6,1,0,9,0,0,0,6,3,5,7,0,0,0,3,0,0,8,1,0,4,3,0,0,10,1,0,0,0
0,6,0,1,9,0,2,8,2,0,6,7,6,1,3,0,3,4,2,0,2,0,0,0,9,8,0,10,4
0,9,9,1,0,0,7,0,0,0,5,10,0,8,0,0,0,5,3,0,0,8,0,7,10,0,0,0
2,8,0,0,0,1,1,8,0,3,4,0,0,0,6,0,0,9,8,10,0,0,2,0,6,10,1
4,0,0,2,0,1,1,0,0,0,0,0,0,0,9,9,3,1,3,0,0,0,10,0,7,5,10
0,4,0,8,7,8,0,1,2,7,4,9,0,0,5,0,8,3,0,0,6,5,0,3,10,3,1,5,0,5
0,3,6,2,0,0,0,2,1,0,2,0,3,0,9,0,10,0,7,7,8,0,0,0,0,3,3,0,9,3
1,0,3,0,0,3,0,7,0,1,3,7,0,7,3,0,0,0,10,0,9,0,0,5,0,7,0,1,0,7
2,9,5,6,0,4,0,4,2,3,1,8,0,8,0,4,0,0,3,0,0,9,8,0,0,9,1,0,0
0,4,7,7,5,0,0,9,0,7,8,1,2,0,4,0,9,5,3,7,4,0,0,5,0,3,9,4,0,5
2,6,0,6,10,0,0,0,3,0,0,2,1,6,6,9,0,0,0,2,8,0,10,6,2,6,0,0,2,1
8,0,0,1,0,0,0,0,7,8,0,6,1,4,4,0,7,7,9,2,0,2,0,9,0,1,0,0,0
8,10,0,3,0,0,0,5,9,3,0,4,6,4,1,0,10,0,0,0,0,0,0,0,6,10,2,6,5
5,0,3,0,8,6,0,0,0,0,4,0,9,4,0,1,7,0,4,0,0,0,0,8,6,10,7,6,0,5
0,3,0,3,0,0,9,8,10,0,0,9,0,0,10,7,1,3,0,0,6,0,0,7,2,0,4,0,0,4
3,0,0,4,0,0,9,3,0,0,0,5,0,7,0,0,3,1,10,10,0,1,0,0,2,0,0,3,0,6
7,0,8,2,0,9,9,0,7,10,3,3,0,7,0,4,0,10,1,0,6,2,0,0,8,2,0,0,0,7
8,10,1,0,5,0,3,0,7,0,0,7,2,9,0,0,0,10,0,1,9,2,2,1,0,5,6,0,2,7
0,8,0,2,3,8,1,6,8,9,0,4,8,2,0,0,6,0,6,9,1,0,0,0,0,1,1,3,9,2
0,7,4,0,0,10,3,5,0,0,0,0,0,0,0,0,1,2,2,0,1,2,0,1,0,0,10,0,9
10,4,3,0,0,0,0,0,0,9,0,10,2,0,0,0,0,0,2,0,2,1,8,0,2,10,0,4,0
9,4,0,0,8,0,0,3,0,5,8,5,6,0,0,8,7,0,0,1,0,0,8,1,8,0,10,0,9
6,0,0,0,2,0,10,0,0,0,0,2,9,0,6,2,2,8,0,0,1,0,8,1,0,0,0,3,7
5,0,10,9,7,0,10,3,3,7,0,3,6,0,6,10,0,0,2,5,1,0,2,0,0,1,10,0,2,10
1,0,1,8,10,0,0,1,3,0,9,9,0,1,10,7,4,0,0,6,1,0,10,10,0,10,1,0,4,0
1,10,0,0,0,6,7,5,0,1,1,4,0,0,2,6,0,3,0,0,3,10,0,0,0,0,0,1,5,0
5,0,0,10,0,10,5,0,9,0,0,0,2,0,6,0,0,0,0,2,9,0,4,0,3,2,4,5,1,0
0,0,0,4,0,1,10,5,3,7,0,5,1,0,5,5,4,6,7,7,2,9,0,9,7,10,0,0,0,1
```

Örnek data.txt

#### Yöntem:

Öncelikle txt dosyasından ilk satır alınır. Kaç adet virgül varsa boyutu = +1 olarak bulunur.

Matrix'in indisleriyle daha kolay işlem yapmak için matrix txt dosyasından okunur.

Köşegenler ve alt matris ile işlem yapılmaz, matrisin sadece üstü ile işlem yaparız.

Matris'in indislerinde dönerek 0 dışındaki tüm değerler edgeArray e kayıt edilir.

EdgeArray küçükten büyüğe olacak şekilde sıralanır.

ClusterArray(nodeları kümelere ayırdığımız dizi), edgeArray'deki değerleri dolaşarak kullanıcından alınan küme sayısı kadar nodeları kümeye alır.

DistanceArray(kümeler arası en kısa mesafeyi sakladığımız dizi), edgeArray'deki değerleri dolaşır, node'lar eğer clusterArray'de aynı kümedeler ise işlem yapılmaz, farklı kümede iseler distanceArray'e daha önce o kümeler eklenmemişse eklenir.

DistanceArray yazdırılır.

### Uygulama:

Cluster Size = 2

1 0 1 2 3

0 1 6 2 3

1 6 1 0 2

2 2 0 1 5

3 3 2 5 1

Matrix Size = 5

[0 2] = 1

[0 3] = 2

[0 4] = 3

[1 2] = 6

[1 3] = 2

[1 4] = 3

[2 4] = 2

[3 4] = 5

---

[0 2] = 1

[0 3] = 2

[1 3] = 2

[2 4] = 2

[0 4] = 3

[1 4] = 3

[3 4] = 5

[1 2] = 6

---Total Edge = 8---

0 1 2 3 4

0 0 0 0 0

---

0 1 2 3 4

1 1 1 1 5

---

Nodes = (2 4), Clusters = {1 5}, Distance = 2

Distance Array

[1 5] = 2

---

Cluster Size = 3

1 1 0 2 4 1

1 1 3 1 0 2

0 3 1 6 5 3

2 1 6 1 2 0

4 0 5 2 1 2

1 2 3 0 2 1

Matrix Size = 6

[0 1] = 1

[0 3] = 2

[0 4] = 4

[0 5] = 1

[1 2] = 3

[1 3] = 1

[1 5] = 2

[2 3] = 6

[2 4] = 5

[2 5] = 3

[3 4] = 2

[4 5] = 2

---

[0 1] = 1

[0 5] = 1

[1 3] = 1

[0 3] = 2

[1 5] = 2

[3 4] = 2

[4 5] = 2

[1 2] = 3

[2 5] = 3

[0 4] = 4

[2 4] = 5

[2 3] = 6

---Total Edge = 12---

0 1 2 3 4 5

0 0 0 0 0 0

---

0 1 2 3 4 5

1 1 3 1 5 1

---

Nodes = (3 4), Clusters = {1 5}, Distance = 2

Nodes = (1 2), Clusters = {1 3}, Distance = 3

Nodes = (2 4), Clusters = {3 5}, Distance = 5

Distance Array

[1 5] = 2

[1 3] = 3

[3 5] = 5

### Sonuç (hoca tarafından verilen data.txt):

Küme Sayısı = 3 iken;

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
6	6	6	6	5	6	6	6	6	6	6	6	6	6	6	16	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Nodes = (2 15), Clusters = {6 16}, Distance = 3

Nodes = (4 20), Clusters = {5 6}, Distance = 3

Nodes = (4 15), Clusters = {5 16}, Distance = 8

Distance Array

[6 16] = 3

[5 6] = 3

[5 16] = 8

Küme Sayısı = 5 iken;

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
6	2	6	6	5	6	6	6	6	6	6	6	6	6	6	16	17	6	6	6	6	6	6	6	6	6	6	6	6	6

Nodes = (16 24), Clusters = {17 6}, Distance = 2

Nodes = (1 8), Clusters = {2 6}, Distance = 3

Nodes = (1 16), Clusters = {2 17}, Distance = 3

Nodes = (2 15), Clusters = {6 16}, Distance = 3

Nodes = (4 20), Clusters = {5 6}, Distance = 3

Nodes = (15 16), Clusters = {16 17}, Distance = 7

Nodes = (4 15), Clusters = {5 16}, Distance = 8

Nodes = (1 4), Clusters = {2 5}, Distance = 9

Distance Array

[17 6] = 2

[2 6] = 3

[2 17] = 3

[6 16] = 3

[5 6] = 3

[16 17] = 7

[5 16] = 8

[2 5] = 9

Küme Sayısı = 7 iken;

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
6	2	6	6	5	6	6	6	6	6	6	6	6	6	15	16	17	6	6	6	6	6	23	6	6	6	6	6	6	6

Nodes = (13 22), Clusters = {6 23}, Distance = 2  
Nodes = (14 27), Clusters = {15 6}, Distance = 2  
Nodes = (16 24), Clusters = {17 6}, Distance = 2  
Nodes = (1 8), Clusters = {2 6}, Distance = 3  
Nodes = (1 16), Clusters = {2 17}, Distance = 3  
Nodes = (2 15), Clusters = {6 16}, Distance = 3  
Nodes = (4 20), Clusters = {5 6}, Distance = 3  
Nodes = (1 22), Clusters = {2 23}, Distance = 4  
Nodes = (15 16), Clusters = {16 17}, Distance = 7  
Nodes = (4 15), Clusters = {5 16}, Distance = 8  
Nodes = (1 4), Clusters = {2 5}, Distance = 9  
Nodes = (1 14), Clusters = {2 15}, Distance = 10  
Nodes = (14 16), Clusters = {15 17}, Distance = 10

Distance Array

[6 23] = 2  
[15 6] = 2  
[17 6] = 2  
[2 6] = 3  
[2 17] = 3  
[6 16] = 3  
[5 6] = 3  
[2 23] = 4  
[16 17] = 7  
[5 16] = 8  
[2 5] = 9  
[2 15] = 10  
[15 17] = 10