CENG 481 GRAF TEORİ VE UYGULAMALARI
Hafta 12

Prof. Dr. Tufan TURACI
tturaci@pau.edu.tr

Hafta 12
Konular

1-Graflarda Eccentricity Değerine Bağlı Topolojiksel İndeksler

2-Graflarda Derece Değerine Bağlı Topolojiksel İndeksler

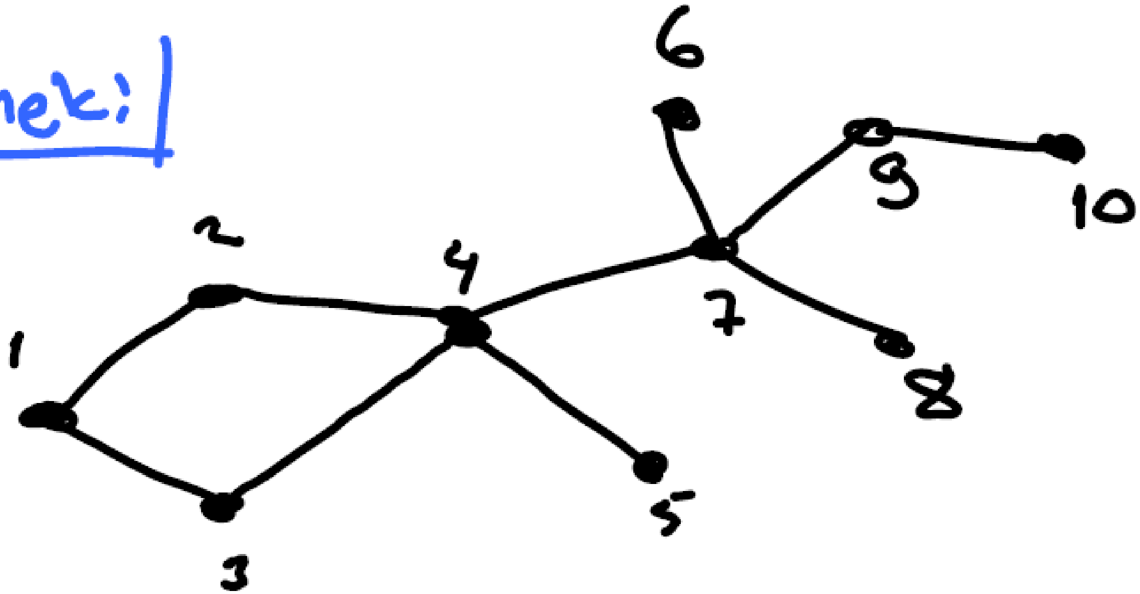
Tanım 1: Bir *G* grafında *u* ve *v* gibi iki tepe arasındaki yollar içinde minimum uzunluğu olanın uzunluğuna; *u* ve *v* nin **uzaklığı (distance)** denir ve *d(u,v)* (veya $d_G(u,v)$) biçiminde gösterilir.

Tanım 2: *n* tepeli bir *G* grafında, grafın tepeleri $\{v_1, v_2, \ldots.., v_n\}$ olsun. *G* grafının **komşuluk matrisi** $A(G) = \left[a_{ij} \right]$ dir.

$$A(G) = \begin{cases} a_{ij} = 1, & v_i \ \ ile \ \ v_j \ \ komşu \ \ ise; \\ a_{ij} = 0, & aksi \ \ halde. \end{cases}$$

$A(G) = \left[a_{ij} \right]$ nin satır ve sütunları grafın tepelerine karşılık gelir.

Örnek:



G

Verilen G grafında tüm tepe çiftleri arasındaki uzaklıkları bulunuz.

$d(1,2) = 1$

$d(1,3) = 1$

$d(1,4) = 2$

$d(1,5) = 3$

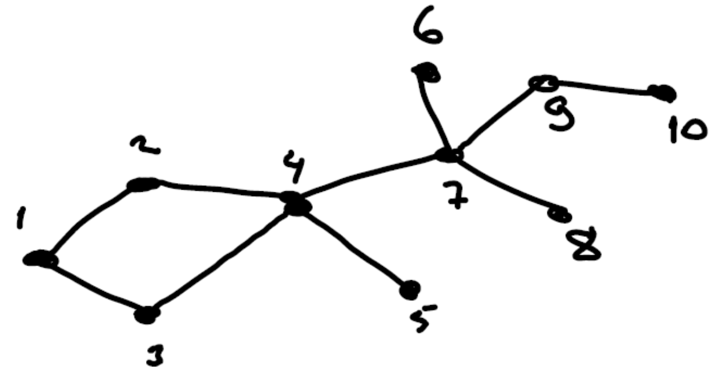
$d(1,6) = 4$

$d(1,7) = 3$

$d(1,8) = 4$

$d(1,9) = 4$

$d(1,10) = 5$



G

$d(2,3) = 2$

$d(2,4) = 1$

$d(2,5) = 2$

$d(2,6) = 3$

$d(2,7) = 2$

$d(2,8) = 3$

$d(2,9) = 3$

$d(2,10) = 4$

$d(3,4) = 1$

$d(3,5) = 2$

$d(3,6) = 3$

$d(3,7) = 2$

$d(3,8) = 3$

$d(3,9) = 3$

$d(3,10) = 4$

$d(4,5) = 1$

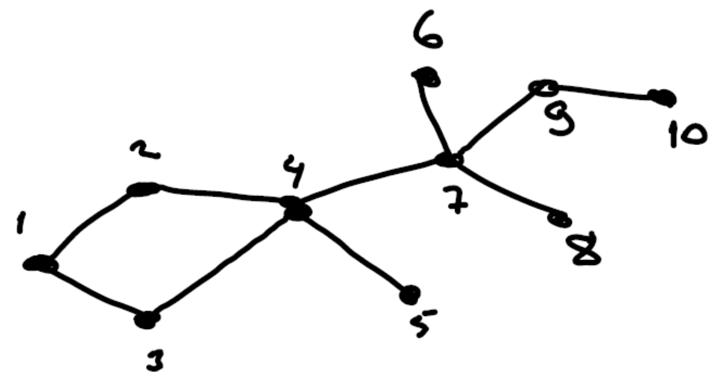
$d(4,6) = 2$

$d(4,7) = 1$

$d(4,8) = 2$

$d(4,9) = 2$

$d(4,10) = 3$



G

$d(5,6) = 3$

$d(5,7) = 2$

$d(5,8) = 3$

$d(5,9) = 3$

$d(5,10) = 4$

$d(7,8) = 1$

$d(7,9) = 1$

$d(7,10) = 2$

$d(8,9) = 2$

$d(8,10) = 3$

$d(9,10) = 1$

$d(6,7) = 1$

$d(6,8) = 2$

$d(6,9) = 2$

$d(6,10) = 3$

Floyd Algoritması

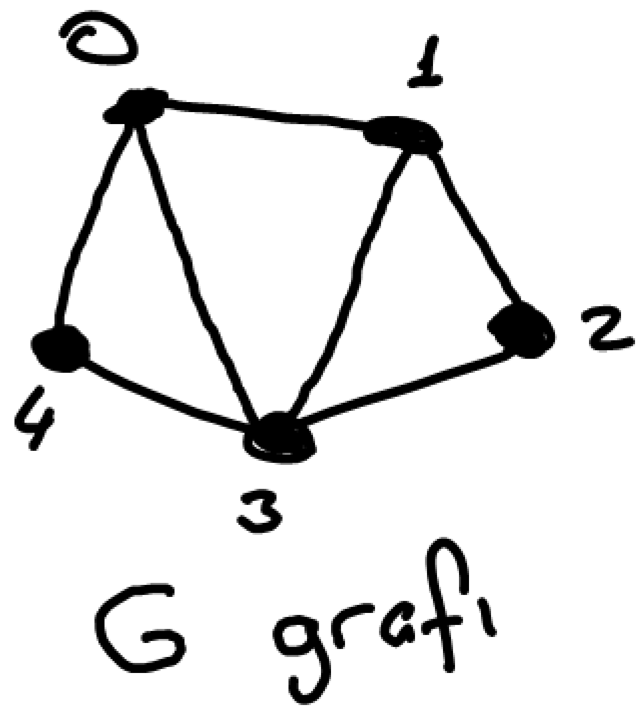
Floyd Algoritması, graf üzerindeki herbir tepe için diğer tepelere olan en kısa yolları ve bu yolların uzaklıklarını bulmak için kullanılan bir algoritmadır. En kısa yolu bulmak için en genel algoritma Floyd'un algoritmasıdır. Grafın bitişiklik matrisi şeklinde tutulması durumunda bu algoritma $O(n^3)$ karmaşıklığında olmaktadır.

Algoritma:

```
for i=1 to n
 for j=1 to n
  if A[i,j] ≠ 0 then D[i,j]=A[i,j]
                    else D[i,j]=∞
 repeat
repeat

for k=1 to n
 for i=1 to n
  for j=1 to n
   if (D[i,k]+D[k,j] < D[i,j]) then D[i,j]=D[i,k]+D[k,j]
  repeat
 repeat
repeat
```

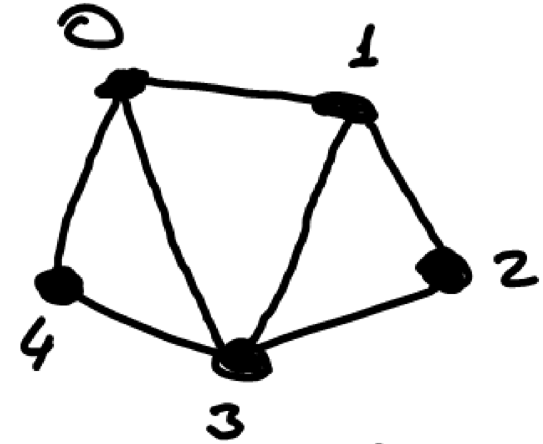
C kodu: Yanda verilen G grafı için...

```c
#include <stdio.h>
#include <conio.h>
#define MAX 1000
 int main()
{
   int i, j, k,n=5;  int D[5][5];
   int A[5][5] = {{0,    1,   0,  1,   1},
                  {1,    0,   1,    1,   0},
                  {0,  1,   0,    1,   0},
                  {1,    1,   1,    0,   1},
                  {1,   0,  0,  1,   0}};

  for (i = 0; i < n; i++){
     for (j = 0; j < n; j++) {
       if (A[i][j]!=0 ) D[i][j] = A[i][j];
       if  (i==j) D[i][j] = 0;
       if ((i!=j) && A[i][j]==0 ) D[i][j] = MAX;

   }}
```



```
for (k = 0; k < n; k++) {
      for (i = 0; i < n; i++){
         for (j = 0; j < n; j++){
            if ((D[i][k] + D[k][j]) < D[i][j])  D[i][j]=D[i][k] + D[k][j];

         }
                  }
                 }

   printf("Tepe ciftleri arasinda en kisa yollar:\n\n");

   for (i = 0; i < n; i++) {
      for (j = 0; j < n; j++) {
      if (i==j) printf("%d - %d  uzaklik = %d ", i,j,0);
                   if (i!=j) printf("%d - %d  uzaklik = %d ", i,j,D[i][j]);
      printf("\n");
      }
   }

   getch();
   return 0;
}
```

```
Tepe ciftleri arasinda en kisa yollar:

0 - 0  uzaklik = 0
0 - 1  uzaklik = 1
0 - 2  uzaklik = 2
0 - 3  uzaklik = 1
0 - 4  uzaklik = 1
1 - 0  uzaklik = 1
1 - 1  uzaklik = 0
1 - 2  uzaklik = 1
1 - 3  uzaklik = 1
1 - 4  uzaklik = 2
2 - 0  uzaklik = 2
2 - 1  uzaklik = 1
2 - 2  uzaklik = 0
2 - 3  uzaklik = 1
2 - 4  uzaklik = 2
3 - 0  uzaklik = 1
3 - 1  uzaklik = 1
3 - 2  uzaklik = 1
3 - 3  uzaklik = 0
3 - 4  uzaklik = 1
4 - 0  uzaklik = 1
4 - 1  uzaklik = 2
4 - 2  uzaklik = 2
4 - 3  uzaklik = 1
4 - 4  uzaklik = 0

--------------------------------
Process exited with return value 0
```



G grafi

Tanım 3: Dışmerkezlilik **(eccentricity)**, her tepenin diğer tepelere olan uzaklıklarının en büyük değeridir ve $e(v)$ (veya $e_G(v)$) biçiminde gösterilir. En büyük dışmerkezlilik değerine **çap (diameter)** denir, $diam(G)$ biçiminde gösterilir. En küçük dışmerkezlilik değerine **yarıçap (radius)** denir ve $r(G)$ biçiminde gösterilir.

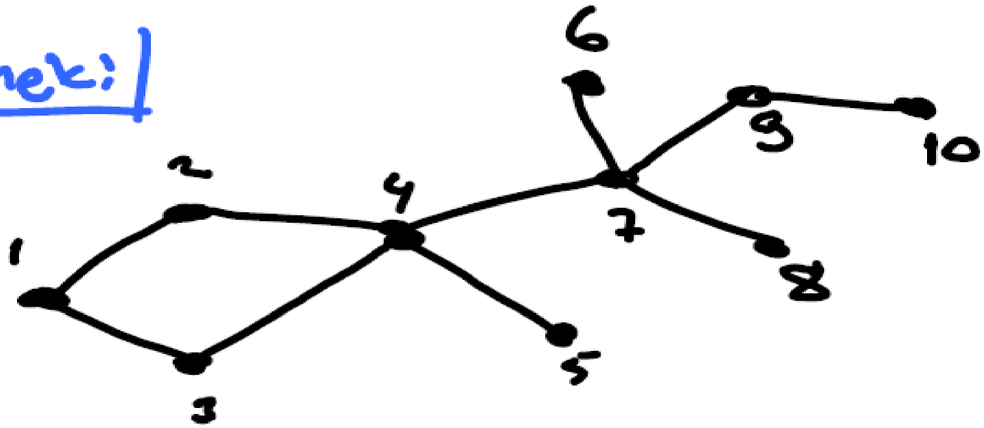
Tanım 4: Dışmerkezlilik değeri yarıçapa eşit olan tepelere **merkez tepeler (central vertices)** denir.

Tanım 5: Dışmerkezlilik değeri çapa eşit olan tepe veya tepelere **kıyı tepeler (peribheral vertices)** denir.

Eccentricity (Dışmerkezlik)

$$e(v) = \max\{\, d(u,v) \mid u,v \in V(G) \,\}$$

Örnek:



G

$e(1) = 5$

$e(2) = 4$

$e(3) = 4$

$e(4) = 3$

$e(5) = 4$

$e(6) = 4$

$e(7) = 3$

$e(8) = 4$

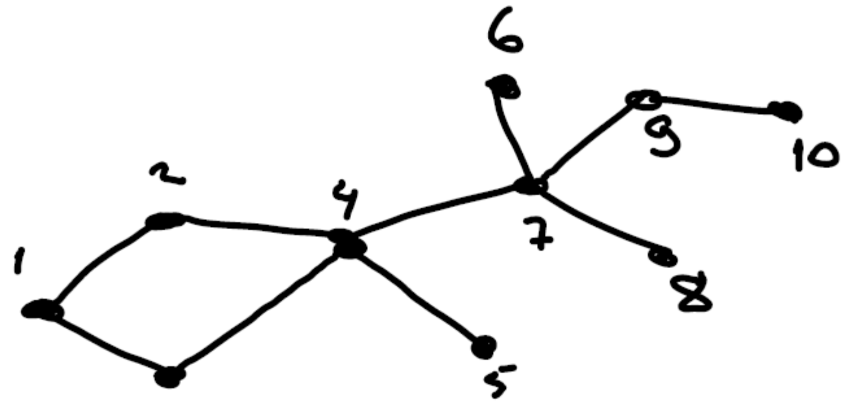
$e(9) = 4$

$e(10) = 5$

Böylece;

$\mathrm{diam}(G) = 5$ (Çap)

$r(G) = 3$ (Yarı çap)



G

Merkez tepeler;

$C(G) = \{ v \mid e(u) = r(G) \}$

$C(G) = \{ 4, 7 \}$

Kıyı tepeler;

$P(G) = \{ v \mid e(u) = \mathrm{diam}(G) \}$

$P(G) = \{ 1, 10 \}$

C kodu: **Eccentricity için yukarıdaki koda aşağıdaki kod parçası eklenir.**

```
for (i=0;i<n;i++)
         {
         for (j=0;j<n;j++)
          {
           if (j==0)  enb=d[i][j];
           if ( d[i][j] >= enb )  enb=d[i][j];
          }
          e[i][0]=enb;
```

```
printf ("\n");
printf  ("Tum tepelerin eccenricity degerleri\n");


for (i=0;i<n;i++)
         {
         for (j=0;j<1;j++)
          {
           printf ("%d--->  %d",i+1, e[i][j]);

}printf ("\n");}
```

C kodu: Floyd Algoritması yardımıyla eccentricity hesabı.

```
#include <stdio.h>
#include <conio.h>
#define MAX 1000
 int main()
{
   int i, j, k,n=5, enb, D[5][5],E[5][5];
   int A[5][5] = {{0,   1,  0,  1,   1},
                  {1,   0,  1,   1,  0},
                  {0,  1,  0,   1,  0},
                  {1,   1,  1,   0,  1},
                  {1,  0,  0,  1,  0}};

  for (i = 0; i < n; i++){
     for (j = 0; j < n; j++) {
       if (A[i][j]!=0 ) D[i][j] = A[i][j];
       if  (i==j) D[i][j] = 0;
       if ((i!=j) && A[i][j]==0 ) D[i][j] = MAX;

   }}
```

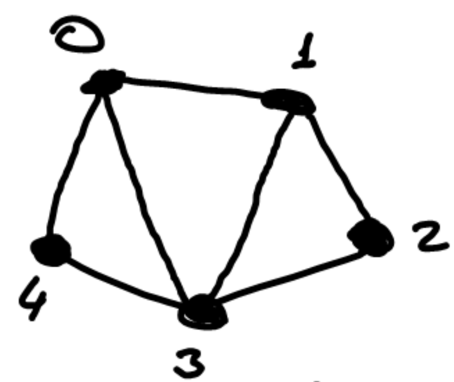
```
for (k = 0; k < n; k++) {
    for (i = 0; i < n; i++){
      for (j = 0; j < n; j++){
        if ((D[i][k] + D[k][j]) < D[i][j])  D[i][j]=D[i][k] + D[k][j];

      }
            }
            }

  printf("Tepe ciftleri arasinda en kisa yollar:\n\n");

  for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
    if (i==j) printf("%d - %d  uzaklik = %d ", i,j,0);
              if (i!=j) printf("%d - %d  uzaklik = %d ", i,j,D[i][j]);
    printf("\n");
    }
  }

printf ("\n");
```

```
printf ( "G nin uzaklik matrisi\n");

for (i=0;i<n;i++)
         {
         for (j=0;j<n;j++)
          {
           if (D[i][j]==MAX) printf ("-");
                              else
                               printf ("%d",D[i][j]);
}printf( "\n" );}

for (i=0;i<n;i++)
         {
         for (j=0;j<n;j++)
          {
           if (j==0)  enb=D[i][j];
           if ( D[i][j] >= enb )  enb=D[i][j];
          }
          E[i][0]=enb;
}

printf ("\n");
```

```
printf  ("Tum tepelerin eccenricity degerleri\n");


for (i=0;i<n;i++)
        {
        for (j=0;j<1;j++)
         {
          printf ("%d--->  %d",i, E[i][j]);

}printf ("\n");}

   getch();
   return 0;
}
```

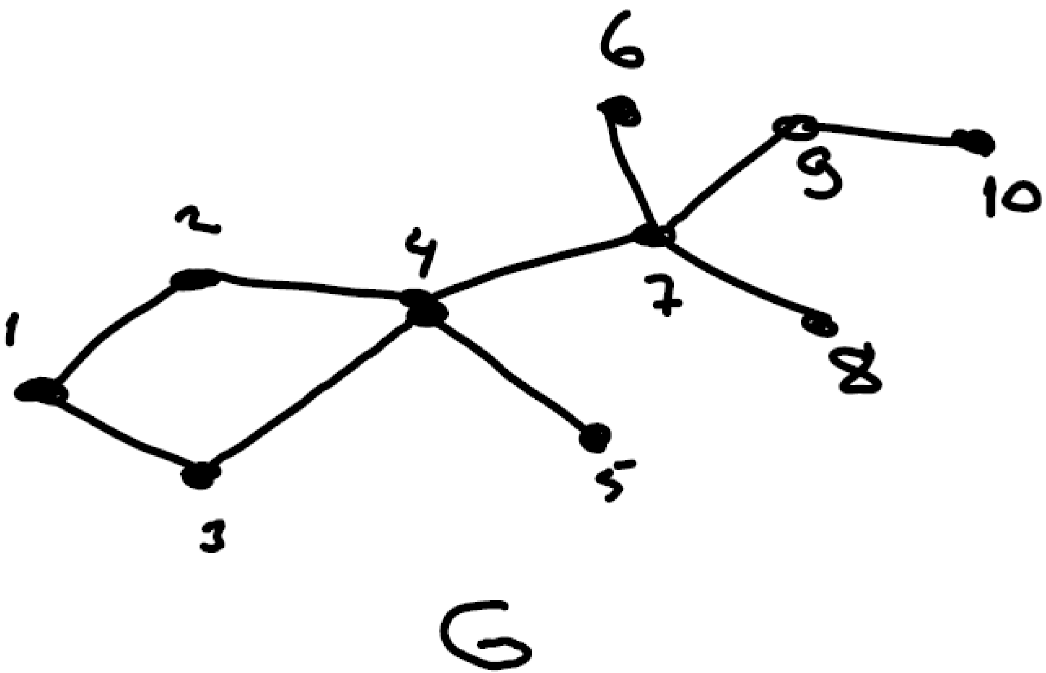
```
Tepe ciftleri arasinda en kisa yollar:

0 - 0  uzaklik = 0
0 - 1  uzaklik = 1
0 - 2  uzaklik = 2
0 - 3  uzaklik = 1
0 - 4  uzaklik = 1
1 - 0  uzaklik = 1
1 - 1  uzaklik = 0
1 - 2  uzaklik = 1
1 - 3  uzaklik = 1
1 - 4  uzaklik = 2
2 - 0  uzaklik = 2
2 - 1  uzaklik = 1
2 - 2  uzaklik = 0
2 - 3  uzaklik = 1
2 - 4  uzaklik = 2
3 - 0  uzaklik = 1
3 - 1  uzaklik = 1
3 - 2  uzaklik = 1
3 - 3  uzaklik = 0
3 - 4  uzaklik = 1
4 - 0  uzaklik = 1
4 - 1  uzaklik = 2
4 - 2  uzaklik = 2
4 - 3  uzaklik = 1
4 - 4  uzaklik = 0
```

```
G nin uzaklik matrisi
01211
10112
21012
11101
12210

Tum tepelerin eccenricity degerleri
0--->  2
1--->  2
2--->  2
3--->  1
4--->  2


--------------------------------
Process exited with return value 0
Press any key to continue . . .
```



G grafi

Örnek: **Aşağıda verilen G grafı için C kodu; verilen ayrıt listesine göre öncelikle komşuluk matrisi oluşturup, daha sonra eccentricity değerlerini hesaplamaktadır.**



C kodu:

```
#include <stdio.h>
#include <conio.h>
#define MAX 1000
 int main()
{int i, j, k,n=10, enb, D[10][10],E[10][10];
 int A[10][10],t[10][10]={{0,0,1,2,3,3,5,6,6,8},{1,2,3,3,4,6,6,7,8,9}};

for (i=0;i<n;i++)
{
for (j=0;j<n;j++)
 {
      A[i][j]=0;
  }}

for (i=0;i<10;i++)
 {
      A[t[0][i]][t[1][i]]=1;
      A[t[1][i]][t[0][i]]=1;
  }
```

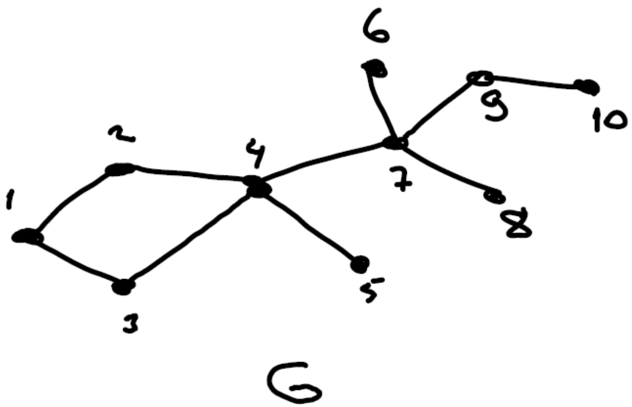
```
printf ("\n");
printf ( "G nin komsuluk matrisi: \n");
for ( i = 0; i <n; i++ ) {
    for ( j = 0; j <n; j++ )
      printf( "%d", A[ i ][ j ] );
      printf( "\n" );}


for (i = 0; i < n; i++){
    for (j = 0; j < n; j++) {
      if (A[i][j]!=0 ) D[i][j] = A[i][j];
      if  (i==j) D[i][j] = 0;
      if ((i!=j) && A[i][j]==0 ) D[i][j] = MAX;
  }}
```

```
for (k = 0; k < n; k++) {
    for (i = 0; i < n; i++){
      for (j = 0; j < n; j++){
        if ((D[i][k] + D[k][j]) < D[i][j])  D[i][j]=D[i][k] + D[k][j];

      }
           }
          }

  printf("Tepe ciftleri arasinda en kisa yollar:\n\n");

  for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
    if (i==j) printf("%d - %d  uzaklik = %d ", i+1,j+1,0);
            if (i!=j) printf("%d - %d  uzaklik = %d ", i+1,j+1,D[i][j]);
    printf("\n");
    }
  }
```

```
printf ("\n");
printf ( "G nin uzaklik matrisi\n");

for (i=0;i<n;i++)
         {
         for (j=0;j<n;j++)
          {
           if (D[i][j]==MAX) printf ("-");
                                  else
                                   printf ("%d",D[i][j]);
           }printf( "\n" );}

for (i=0;i<n;i++)
         {
         for (j=0;j<n;j++)
          {
           if (j==0)  enb=D[i][j];
           if ( D[i][j] >= enb )  enb=D[i][j];
          }
          E[i][0]=enb;
}
```

```
printf ("\n");
printf  ("Tum tepelerin eccenricity degerleri\n");


for (i=0;i<n;i++)
        {
        for (j=0;j<1;j++)
         {
          printf ("%d--->  %d",i+1, E[i][j]);

}printf ("\n");}

   getch();
   return 0;
}
```

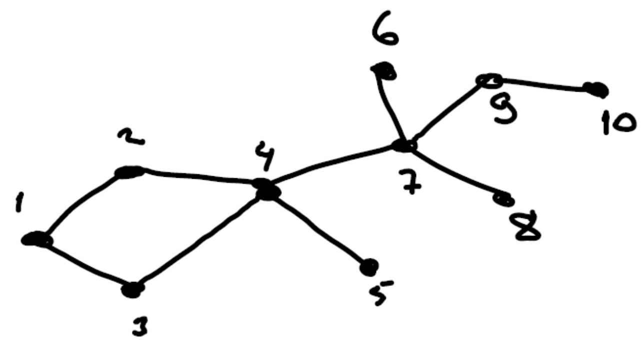
Ekran Çıktısı:




```
G nin komsuluk matrisi:
0110000000
1001000000
1001000000
0110101000
0001000000
0000001000
0001010110
0000001000
0000001001
0000000010
```

```
Tepe ciftleri arasinda en kisa yollar:

1 - 1  uzaklik = 0
1 - 2  uzaklik = 1
1 - 3  uzaklik = 1
1 - 4  uzaklik = 2
1 - 5  uzaklik = 3
1 - 6  uzaklik = 4
1 - 7  uzaklik = 3
1 - 8  uzaklik = 4
1 - 9  uzaklik = 4
1 - 10  uzaklik = 5
2 - 1  uzaklik = 1
2 - 2  uzaklik = 0
2 - 3  uzaklik = 2
2 - 4  uzaklik = 1
2 - 5  uzaklik = 2
2 - 6  uzaklik = 3
2 - 7  uzaklik = 2
2 - 8  uzaklik = 3
2 - 9  uzaklik = 3
2 - 10  uzaklik = 4
```

```
G nin uzaklik matrisi
0112343445
1021232334
1201232334
2110121223
3221032334
4332301223
3221210112
4332321023
4332321201
5443432310

Tum tepelerin eccenricity degerleri
1--->  5
2--->  4
3--->  4
4--->  3
5--->  4
6--->  4
7--->  3
8--->  4
9--->  4
10--->  5

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```



G

$e(1) = 5$
$e(2) = 4$
$e(3) = 4$
$e(4) = 3$

$e(5) = 4$
$e(6) = 4$
$e(7) = 3$
$e(8) = 4$
$e(9) = 4$
$e(10) = 5$

TOPOLOJİKSEL İNDEKSLER

Graf teori, bir moleküler grafik / ağ mimarisinin analizi ve çalışmasında en güçlü matematiksel araçlardan biri haline gelmiştir. Ağlar önemli yapılardır ve birçok farklı uygulama ve ortamda görünür. Ağların incelenmesi, kimya, bilgisayar bilimi, matematik, sosyal bilimler, bilişim ve diğer teorik ve uygulamalı bilimleri içeren çok disiplinli araştırmanın önemli bir alanı haline gelmiştir.

Kimyasal graf teorisi, moleküllerin matematiksel modellemesi ile ilgili olan grafik teorisinin önemli bir dalıdır. Aynı zamanda topolojik indekslerin gelişimini de ele alır.

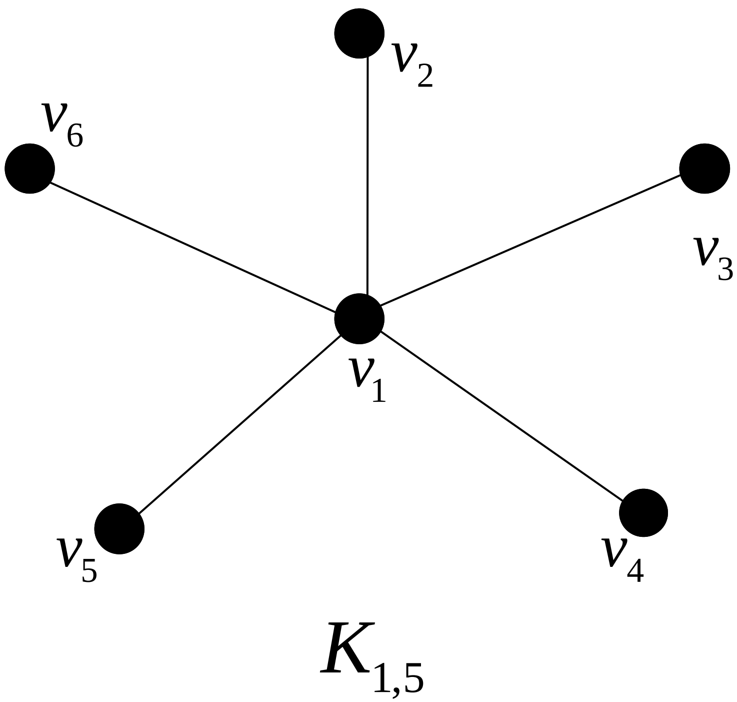
Topolojik indeksler, kimyasal bileşiklerin yapılarını tanımlayan moleküler tanımlayıcılardır ve kaynama noktası, buharlaşma entalpisi ve stabilite gibi belirli fiziko-kimyasal özellikleri tahmin etmemize yardımcı olur.

Ağın sağlamlığını ölçmek için çeşitli önlemler tanımlanmış ve ağ zafiyetini hesaplamak için formüller türetmek için çeşitli grafik teorik parametreler kullanılmıştır.

Wiener indeksi, kimyadaki ilk topolojik indekstir ve kimyager Harold Wiener tarafından da tanımlanmıştır. Wiener indeksi, G grafiğinin her bir köşe çifti arasındaki mesafelerin yarısını toplamayı amaçlamaktadır ve şu şekilde tanımlanır:

$$W(G)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}d_G(v_i,v_j)$$

Örnek 1. $K_{1,5}$ yıldız grafının Wiener indeksini hesaplayınız.



Tüm tepe çiftleri arasındaki uzaklıklar:

$d_{K_{1,5}}(v_1,v_2)=1$ $d_{K_{1,5}}(v_2,v_3)=2$ $d_{K_{1,5}}(v_3,v_4)=2$

$d_{K_{1,5}}(v_1,v_3)=1$ $d_{K_{1,5}}(v_2,v_4)=2$ $d_{K_{1,5}}(v_3,v_5)=2$

$d_{K_{1,5}}(v_1,v_4)=1$ $d_{K_{1,5}}(v_2,v_5)=2$ $d_{K_{1,5}}(v_3,v_6)=2$

$d_{K_{1,5}}(v_1,v_5)=1$ $d_{K_{1,5}}(v_2,v_6)=2$

$d_{K_{1,5}}(v_1,v_6)=1$

$$W(K_{1,5})=\frac{1}{2}\sum_{i=1}^{6}\sum_{j=1}^{6}d_{K_{1,5}}(v_i,v_j)$$

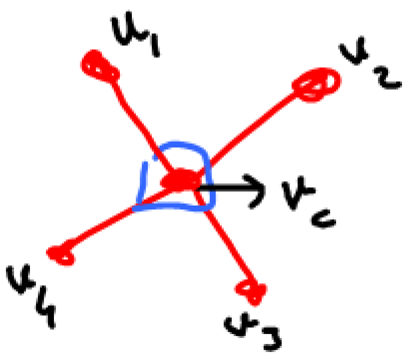
$$W(K_{1,5})=25$$

Örnek 2. $n+1$ tepeli $K_{1,n}$ yıldız grafların Wiener İndeks değerini hesaplayınız.

Wiener index

$$\boxed{W(G)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_G(u_i,u_j)}$$

$K_{1,n}$ grafı için?



$d_G(u_c,u_i)=1$, $i\in\{1,\dots,n\}$

$u_i, u_j \in V(G)-\{u_c\}$ olsun.

$d_G(u_i,u_j)=2$ $\quad i,j\in\{1,\dots,n\}$

$K_{1,5}=25$

Genel Formül:

$$W(G) = \frac{1}{2}\left(n.1 + n.(\overbrace{(n-1).2+1}^{2n-1}) \right)$$

$$w(G) = \frac{1}{2}\left(\cancel{n} + 2n^2 - \cancel{n} \right)$$

$$w(G) = n^2 \Rightarrow \boxed{w(K_{1,n}) = n^2}$$

C kodundaki
vermiş olduğumuz uzaklık matrisinden

sonra :

```
top=0
for {i=0 ; i<n ; i++}
  {
    for (j=0 ; j<n ; j++}
      {
        top=top+d[i][j]
      }
  }

printf ("G grafının wiener index
         değeri =%d , top/2};
```

Çalışma Sorusu: n tepeli yol, çevre, tam ve tekerlek grafların Wiener İndeks değerlerini hesaplayınız.

Eccentricity(Dış Merkezlik) Temelli Topolojiksel İndeksler

The connective eccentricity index: $\xi^{ce}(G) = \sum_{u \in V(G)} \left(\deg_G(u) / \varepsilon_G(u) \right).$

The eccentric connectivity index: $\xi^{c}(G) = \sum_{u \in V(G)} \left(\deg_G(u).\varepsilon_G(u) \right).$

The total eccentricity index: $\xi(G) = \sum_{u \in V(G)} \varepsilon_G(u).$

The first Zagreb eccentricity index : $M_1^*(G) = \sum_{uv \in E(G)} \left(\varepsilon_G(u) + \varepsilon_G(v) \right).$

The second Zagreb eccentricity index: $M_1^{**}(G) = \sum_{u \in V(G)} \left(\varepsilon_G(u) \right)^2.$

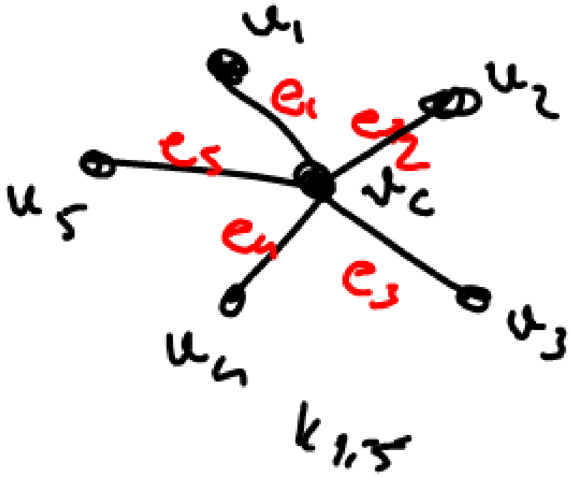
The third Zagreb eccentricity index: $M_2^*(G) = \sum_{uv \in E(G)} \left(\varepsilon_G(u).\varepsilon_G(v) \right).$

The eccentricity based geometric-arithmetic index: $GA_4(G) = \sum_{uv \in E(G)} \left(\frac{2\sqrt{\varepsilon_G(u).\varepsilon_G(v)}}{\varepsilon_G(u) + \varepsilon_G(v)} \right).$

New version of the ABC index namely $ABC_5(G)$: $ABC_5(G) = \sum_{uv \in E(G)} \left(\sqrt{\frac{\varepsilon_G(u) + \varepsilon_G(v) - 2}{\varepsilon_G(u).\varepsilon_G(v)}} \right).$

Örnek)

$K_{1,5}$ grafı için yukarıdaki parametre değerlerini hesaplayınız.



$$\varepsilon(v_c) = 1$$

$$\varepsilon(v_1) = \varepsilon(v_2) = \varepsilon(v_3) = \varepsilon(v_4) = \varepsilon(v_5) = 2$$

1) connective eccentricity index:

$$\xi^{ce}(K_{1,5}) = \sum_{u_i} \left(\frac{\deg(u_i)}{\varepsilon(u_i)} \right)$$

$$= \frac{5}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = 5 + \frac{5}{2} = \boxed{\frac{15}{2}}$$

Bu formül için kod, verilen C programına eklenebilir.

2) Eccentric Connectivity indeks

$$\xi^c(K_{1,5}) = \sum_{u_i} \left(\deg(u_i) \cdot \varepsilon(u_i) \right)$$

$$= 5.1 + 1.2 + 1.2 + 1.2 + 1.2 + 1.2$$

$$\boxed{= 15}$$

3) Total Eccentricity indeks

$$\xi(K_{1,5}) = 1 + 2 + 2 + 2 + 2 + 2 = 11$$

$$7$$

4) 1. Zagreb Eccentricity indeks

$$M_1^*(K_{1,5}) = \sum_{uv \in E(K_{1,5})} \left(\varepsilon(v) + \varepsilon(u) \right)$$

$$= \overbrace{\underset{1}{\varepsilon(u_c)} + \underset{2}{\varepsilon(u_1)}}^{e_1} + \overbrace{\underset{1}{\varepsilon(u_c)} + \underset{2}{\varepsilon(u_2)}}^{e_2}$$

$$\overbrace{\underset{1}{\varepsilon(u_c)} + \underset{2}{\varepsilon(u_3)}}^{e_3} + \overbrace{\underset{1}{\varepsilon(u_c)} + \underset{2}{\varepsilon(u_4)}}^{e_4}$$

$$\overbrace{\underset{1}{\varepsilon(u_c)} + \underset{2}{\varepsilon(u_5)}}^{e_5} = \boxed{15}$$

5) İkinci Zagreb Eccentricity indeks

$$M_1^{**}(K_{1,5}) = 1^2+2^2+2^2+2^2+2^2+2^2 = \underline{21}$$

6) Üçüncü Zagreb Eccentricity indeks

$$M_2^{*}(K_{1,5}) = \sum_{uv \in E(K_{1,5})} (\varepsilon(u) \cdot \varepsilon(v))$$

$$= \Big(\overbrace{\underset{1}{\varepsilon(u_c)} \cdot \underset{2}{\varepsilon(u_1)}}^{e_1}\Big) + \Big(\overbrace{\underset{1}{\varepsilon(u_c)} \cdot \underset{2}{\varepsilon(u_2)}}^{e_2}\Big)$$

$$\Big(\overbrace{\underset{1}{\varepsilon(u_c)} \cdot \underset{2}{\varepsilon(u_3)}}^{e_3}\Big) + \Big(\overbrace{\underset{1}{\varepsilon(u_c)} \cdot \underset{2}{\varepsilon(u_4)}}^{e_4}\Big)$$

$$\Big(\overbrace{\underset{1}{\varepsilon(u_c)} \cdot \underset{2}{\varepsilon(u_5)}}^{e_5}\Big) = \boxed{10}$$

7) Eccentricity temelli: Geometrik-Aritmetik indeks

$$GA_4(K_{1,5}) = \sum_{uv \in E(K_{1,5})} \left(\frac{2\sqrt{\varepsilon(u)\cdot\varepsilon(v)}}{\varepsilon(u)+\varepsilon(v)} \right)$$

$$= \overbrace{\frac{2\sqrt{\varepsilon(u_c)\cdot\varepsilon(u_1)}}{\varepsilon(u_c)+\varepsilon(u_1)}}^{e_1} + \overbrace{\frac{2\sqrt{\varepsilon(u_c)\cdot\varepsilon(u_2)}}{\varepsilon(u_c)+\varepsilon(u_2)}}^{e_2}$$

$$\overbrace{\frac{2\sqrt{\varepsilon(u_c)\cdot\varepsilon(u_3)}}{\varepsilon(u_c)+\varepsilon(u_3)}}^{e_3} + \overbrace{\frac{2\sqrt{\varepsilon(u_c)\cdot\varepsilon(u_4)}}{\varepsilon(u_c)+\varepsilon(u_4)}}^{e_4}$$

$$\overbrace{\frac{2\sqrt{\varepsilon(u_c)\cdot\varepsilon(u_5)}}{\varepsilon(u_c)+\varepsilon(u_5)}}^{e_5} = ?$$

$$= \underbrace{\frac{2\sqrt{1 \cdot 2}}{3} + \frac{2 \cdot \sqrt{1 \cdot 2}}{3} + \cdots + \frac{2\sqrt{1 \cdot 2}}{3}}_{5 \text{ tane var.}}$$

$$= \frac{2\sqrt{2}}{3} \cdot 5 = \boxed{\frac{10\sqrt{2}}{3}}$$

8) Eccentricity Temelli Atom-Bond Connectivity indeksi:

$$ABC_5(K_{1,5}) = \sum_{uv\in E(K_{1,5})} \left(\sqrt{\frac{\varepsilon(u)+\varepsilon(v)-2}{\varepsilon(u)\cdot\varepsilon(v)}}\right)$$

$$ABC(K_{1,5}) = 5\cdot\left(\sqrt{\frac{1+\cancel{2}-\cancel{2}}{1\cdot 2}}\right)$$

$$=5\cdot\sqrt{\frac{1}{2}} = \frac{5}{\sqrt{2}} = \boxed{\frac{5\sqrt{2}}{2}}$$

*) Tüm indeks değerleri verilen C kodu yardımıyla hesaplanabilir.

*) Uzaklık matrisi: (A^k ile bulunan) algoritmasının verimlilik sınıfı: $\boxed{O(n^3)}$

Matris çarpımı

Derece Temelli Topolojiksel İndeksler

The Randic connectivity index: $R(G)=\sum_{uv\in E(G)}\left(\frac{1}{\sqrt{\deg_G(u)\deg_G(v)}}\right).$

The general Randic connectivity index: $R_\alpha(G)=\sum_{uv\in E(G)}\left(\deg_G(u)\deg_G(v)\right)^\alpha.$

The general sum-connectivity index : $X_\alpha(G)=\sum_{uv\in E(G)}\left(\deg_G(u)+\deg_G(v)\right)^\alpha.$

The *first Zagreb index*: $M_1(G)=\sum_{u\in V(G)}\left(\deg_G(u)\right)^2.$

The *second Zagreb index*: $M_2(G)=\sum_{uv\in E(G)}\left(\deg_G(u)\deg_G(u)\right).$

The *harmonic index*: $H(G)=\sum_{uv\in E(G)}\left(\frac{2}{\deg_G(u)+\deg_G(v)}\right).$

The *geometric-arithmetic (GA) index*: $GA(G)=\sum_{uv\in E(G)}\left(\frac{2\sqrt{\deg_G(u)\deg_G(v)}}{\deg_G(u)+\deg_G(v)}\right).$

Komşuluk matrisi:

derece ?

$$A(G) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}} \end{array}$$

$deg(1) = 3$

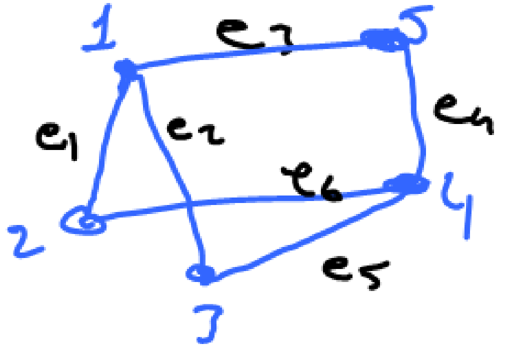
$deg(2) = 2$

$deg(3) = 2$

$deg(4) = 3$

$deg(5) = 2$

satır toplamı.



$\Rightarrow$ G graf,

Tepe	derece
1	3
2	2
3	2
4	3
5	2

1) Randic index:

$$R(G) = \sum_{uv \in E(G)} \frac{1}{\sqrt{\deg(u) \cdot \deg(v)}}$$

$$= \frac{1}{\sqrt{3 \cdot 2}} + \frac{1}{\sqrt{3 \cdot 2}} + \frac{1}{\sqrt{3 \cdot 2}} + \frac{1}{\sqrt{3 \cdot 2}} + \frac{1}{\sqrt{3 \cdot 2}} + \frac{1}{\sqrt{3 \cdot 2}}$$

$$= \frac{6}{\sqrt{6}} = \frac{\cancel{6}\sqrt{6}}{\cancel{6}} = \boxed{\sqrt{6}}$$

ii) Genel Randic index:

$$R_\alpha(G) = \sum_{uv \in E(G)} (\deg(u)\cdot\deg(v))^\alpha$$

$$= 6 \cdot 6^\alpha = \boxed{6^{\alpha+1}}$$

$\alpha = -\frac{1}{2} \Rightarrow$ Randic index $= 6^{-\frac{1}{2}+1} = 6^{\frac{1}{2}} = \sqrt{6}$

3) Genel toplam-connectivity.

$$\chi_\alpha(G) = \sum_{uv \in E(G)} (\deg(u) + \deg(v))^\alpha$$

$$\boxed{\;= 6 \cdot 5^\alpha\;}$$

4) Birinci Zagreb index

$$M_1(G) = \sum_{v \in V(G)} (\deg(v))^2$$

$$M_1(G) = 3^2 + 2^2 + 2^2 + 3^2 + 2^2$$

$$= 18 + 12 = \underline{30}$$

5) ikinci Zagreb indeksi

$$M_2(G) = \sum_{uv \in E(G)} (\deg(u) \cdot \deg(v))$$

$$= 6 \cdot (2 \cdot 3) = \underline{\underline{36}}$$

6) Harmonic index

$$H(G) = \sum_{uv \in E(G)} \frac{2}{\deg(u) + \deg(v)}$$

$$H(G) = \left(\frac{2}{2+3}\right) \cdot 6 = \frac{2}{5} \cdot 6 = \boxed{\frac{12}{5}}$$

7) Geometric-Aritmetik indeksi

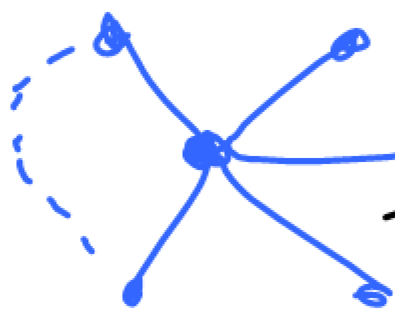
$$GA(G) = \sum_{uv \in E(G)} \frac{2\sqrt{d_G(u) \cdot d_G(v)}}{d_G(u) + d_G(v)}$$

$$GA(G) = 6 \cdot \left(\frac{2\sqrt{2 \cdot 3}}{2+3} \right)$$

$$= 6 \cdot \frac{(2\sqrt{5})}{5} = \boxed{\frac{12}{5}\sqrt{5}}$$

Örnek:

$K_{1,n}$ yıldız grafının derece temelli topolojiksel indeks değerlerini elde ediniz.



$K_{1,n}$

1 tane n dereceli tepe var.

n tane 1 dereceli tepe var.

n tane ayrıt var.

1) Randić indeksi

$$R(K_{1,n}) = \sum_{uv \in E(G)} \frac{1}{\sqrt{\deg(u)\deg(v)}}$$

$$= n \cdot \left(\frac{1}{\sqrt{1 \cdot n}} \right) = \frac{n}{\sqrt{n}} = \sqrt{n}$$

Genel Randić indeksi

$$R_\alpha(K_{1,n}) = \sum_{uv \in E(G)} \left[\deg(u)\cdot\deg(v)\right]^\alpha$$

$$= n.(1.n)^\alpha = \boxed{n^{\alpha+1}}$$

$\boxed{\alpha = -\frac{1}{2} \text{ için } n^{\frac{1}{2}} = \sqrt{n} \text{ 'dir.}}$

3) General [illegible]-connectivity index

$$\chi_\alpha(K_{1,n}) = \sum_{uv \in E(G)} (\deg(u) + \deg(v))^\alpha$$

$$= \boxed{n \cdot (1+n)^\alpha}$$

4) 1. Zagreb index

$$M_1(G) = \sum_{v \in V(G)} (\deg(v))^2$$

$$M_1(K_{1,n}) = 1 \cdot n^2 + n \cdot 1^2$$

$$\boxed{= n^2 + n}$$

5) 2. Zagreb indeks

$$M_2(G) = \sum_{uv \in E(G)} (\deg(u) \cdot \deg(v))$$

$$M_2(K_n) = n \cdot (n-1) = n^2$$

6) Harmonic indeks

$$H(G) = \sum_{uv \in E(G)} \frac{2}{deg(v) + deg(u)}$$

$$H(K_{1,n}) = n \cdot \frac{2}{n+1} = \boxed{\frac{2n}{n+1}}$$

7) Geometrik-Arithmetik index(G)

$$GA(G) = \sum_{uv \in E(G)} \left(\frac{2\sqrt{\deg(u) \cdot \deg(v)}}{\deg(u) + \deg(v)} \right)$$

$$GA(K_{1,n}) = n \cdot \left(\frac{2 \cdot \sqrt{n \cdot 1}}{n+1} \right) = \boxed{\frac{2n\sqrt{n}}{n+1}}$$

* Topolojiksel indeksler

- polinom zamanda hesaplanabilir.

- Özel tipte graflar için indeks değerleri formülü bulunabiliyor.

yol, çevre, tam, tekerlek

*) P_n, C_n, K_n ve $W_{1,n}$ grafların derece temelli topolojiksel indeks değerlerini hesaplayınız.

KAYNAKLAR

[1] Chartrand, G.-Lesniak, L., (1986) : *Graphs and Digraphs*, Wadsworth & Brooks, California

[2] West D.B. (2001) : *Introduction to Graph Theory*, *Prentice Hall*, USA.

[3] Graf Teoriye Giriş, Şerife Büyükköse ve Gülistan Kaya Gök, Nobel Yayıncılık

[4] Discrete Mathematical Structures for Computer Science, Ronald E. Prather, Houghton Mifflin Company, (1976).

[5] Christofides, N., 1986. Graph Theory an Algorithmic Approach, Academic Press, London

[6] ALGORİTMALAR (Teoriden Uygulamalara), Vasif V. NABİYEV, Seçkin Yayıncılık