CENG 114 BİLGİSAYAR BİLİMLERİ İÇİN AYRIK YAPILAR

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

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü

- Hafta 14

Ders İçeriği

- **Graf Veri Yapısı**
- **Graflarda Uzaklık Kavramı**

--- Bir grafın çapı, yarıçapı, merkez ve kıyı tepeleri

--- Dijsktra Algoritması

- **Graflarda Bağlantılılık (Graph Connectivity)**

Graf Veri Yapısı

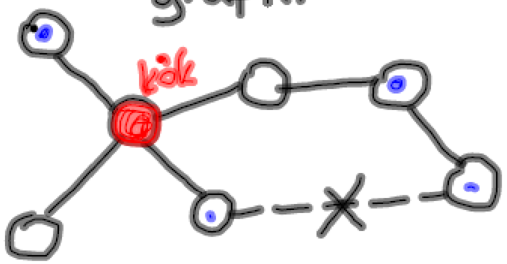
--- Bilgisayar dünyasında bulunan ve gerçek hayatta çeşitli sebeplerle karşılaşılan yapıları temsil etmek amacıyla graflar kullanılır.

--- Örneğin bir bilgisayar ağını, her türlü iletişim problemini graflar ile modelleyebiliriz veya küçükten büyüğe sıralanmış sayıları saklamak için ağaç grafları kullanabiliriz.

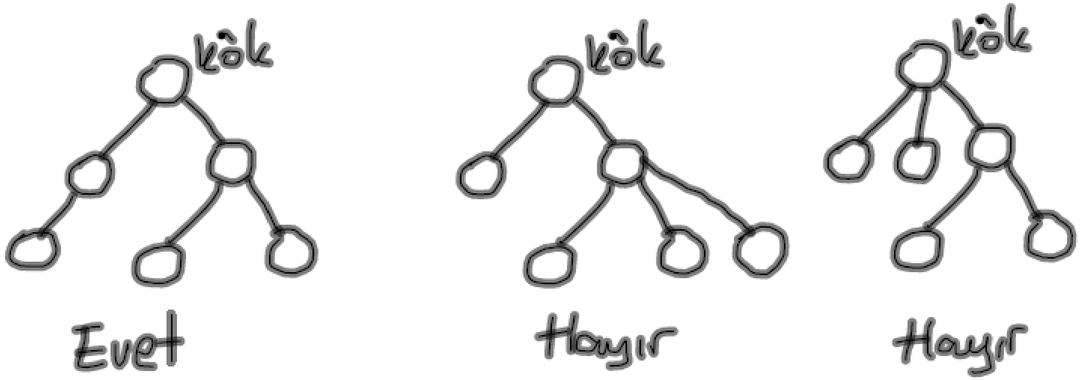
--- Belirli bir kurala göre bir ikili ağaçta dolaşma ve arama için graf veri yapısı kullanılır.

Örnek: İkili Arama Ağacı Binary Search Tree (BST)

Ağaç: Döngü (çevre, çevrim, cycle) içermeyen bir graf
veya Her iki düğümü arasında tam 1 yol içeren graftır.



İkili ağaç: (Binary tree): Her düğümün en fazla 2 çocuğu olabilir.



İkili Arama Ağacı (Binary Search Tree):

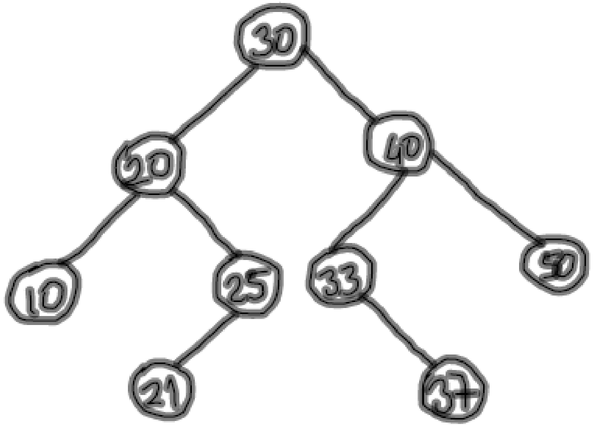
x ikili arama ağacında bir düğüm olsun.

Eğer y düğümü x'in sol alt ağacında ise

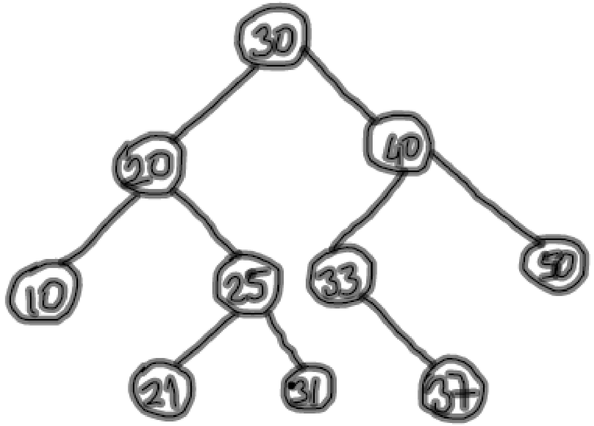
$$key[y] \leq key[x]$$

aksi durumda ise

$$key[y] > key[x]$$



BST dir



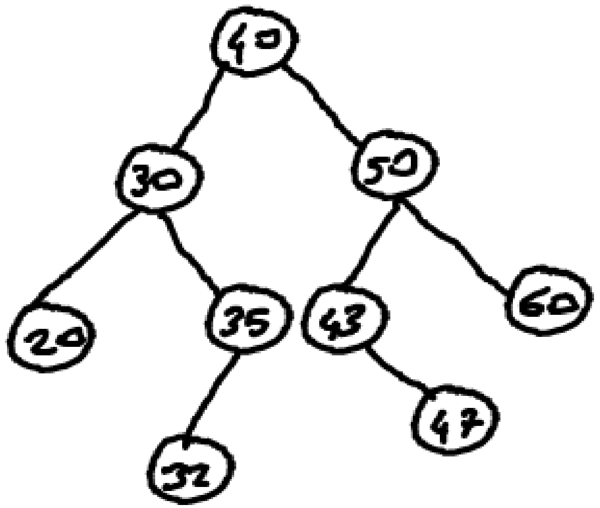
BST değil

İkili Ağaçlarda Dolaşma

Inorder Tree walk: sol - kök - sağ

Preorder Tree walk: kök - sol - sağ

Postorder Tree walk: sol - sağ - kök



Inorder: 20, 30, 32, 35, 40, 43, 47, 50, 60

Preorder: 40, 30, 20, 35, 32, 50, 43, 47, 60

postorder: 20, 32, 35, 30, 47, 43, 60, 50, 40

Graflarda Uzaklık Kavramı

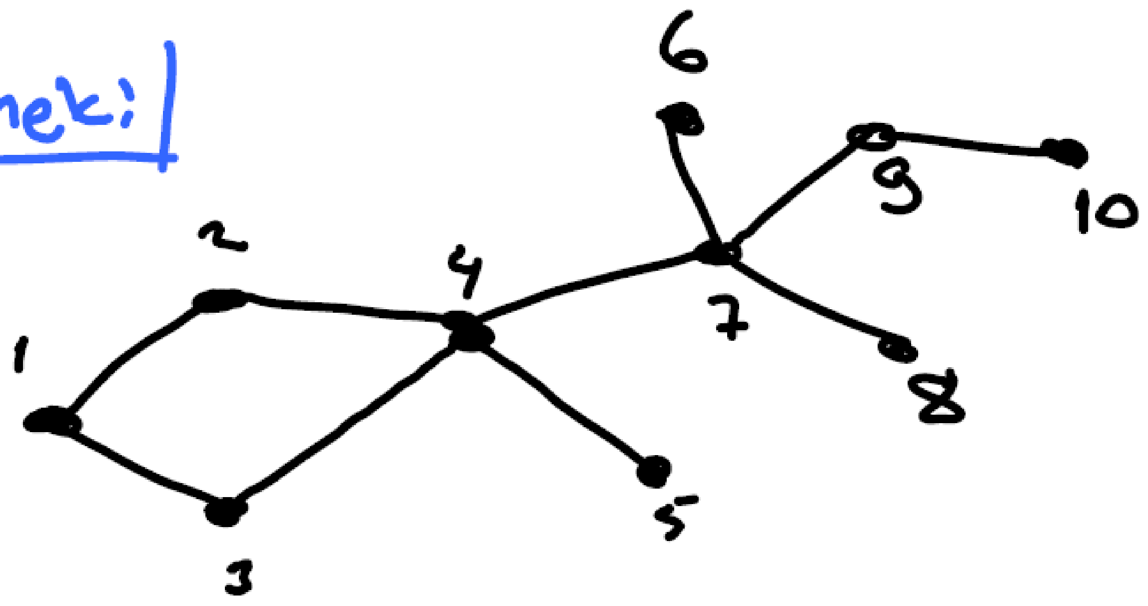
Tanım 1: Bir G grafında u ve v gibi iki tepe arasındaki yollar içinde minimum uzunluğu olanın uzunluğuna; u ve v nin **uzaklığı (distance)** denir ve $d(u,v)$ (veya $d_G(u,v)$) biçiminde gösterilir.

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.

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

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

Örnek:



G

Verilen G grafında tüm tepe çiftleri arasındaki uzaklıkları bulunuz.

$d(1,2) = 1$

$d(1,3) = 1$

$d(1,4) = 2$

$d(1,5) = 3$

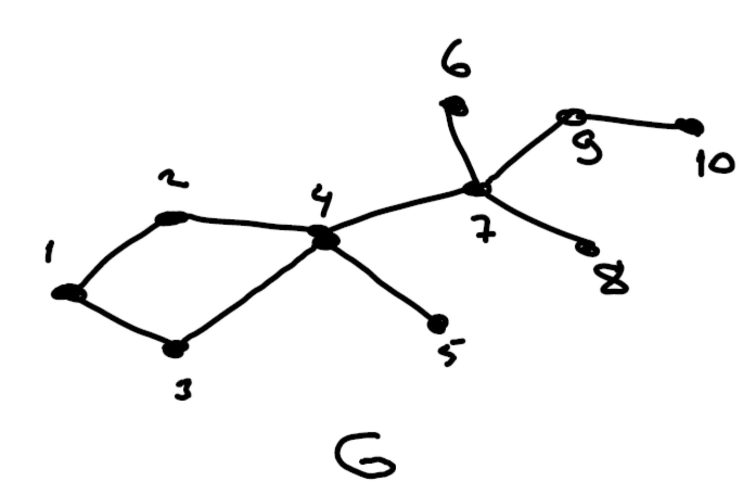
$d(1,6) = 4$

$d(1,7) = 3$

$d(1,8) = 4$

$d(1,9) = 4$

$d(1,10) = 5$



$d(2,3) = 2$

$d(2,4) = 1$

$d(2,5) = 2$

$d(2,6) = 3$

$d(2,7) = 2$

$d(2,8) = 3$

$d(2,9) = 3$

$d(2,10) = 4$

$d(3,4) = 1$

$d(3,5) = 2$

$d(3,6) = 3$

$d(3,7) = 2$

$d(3,8) = 3$

$d(3,9) = 3$

$d(3,10) = 4$

$d(4,5) = 1$

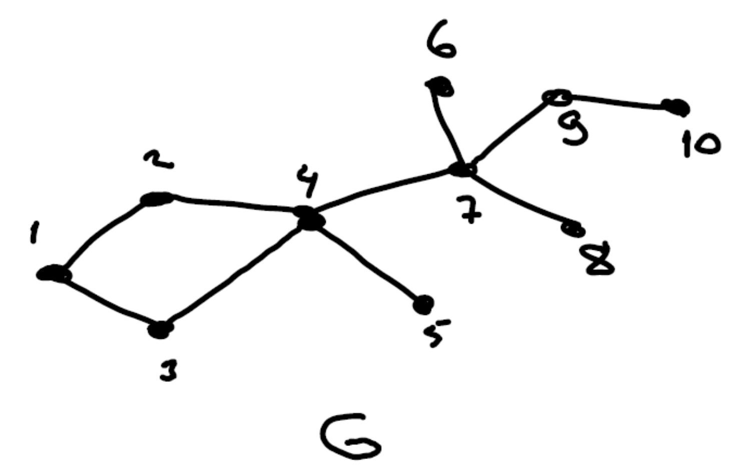
$d(4,6) = 2$

$d(4,7) = 1$

$d(4,8) = 2$

$d(4,9) = 2$

$d(4,10) = 3$



$d(5,6) = 3$

$d(5,7) = 2$

$d(5,8) = 3$

$d(5,9) = 3$

$d(5,10) = 4$

$d(6,7) = 1$

$d(6,8) = 2$

$d(6,9) = 2$

$d(6,10) = 3$

$d(7,8) = 1$

$d(7,9) = 1$

$d(7,10) = 2$

$d(8,9) = 2$

$d(8,10) = 3$

$d(9,10) = 1$

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.

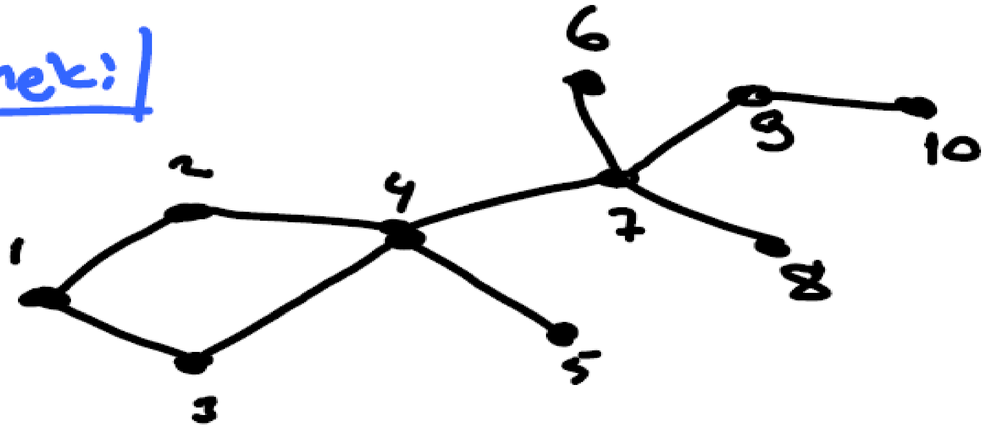
Tanım 4: Dışmerkezlilik değeri yarıçapa eşit olan tepelere **merkez tepeler (central vertices)** denir.

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

Eccentricity (Dışmerkezlik)

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

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;

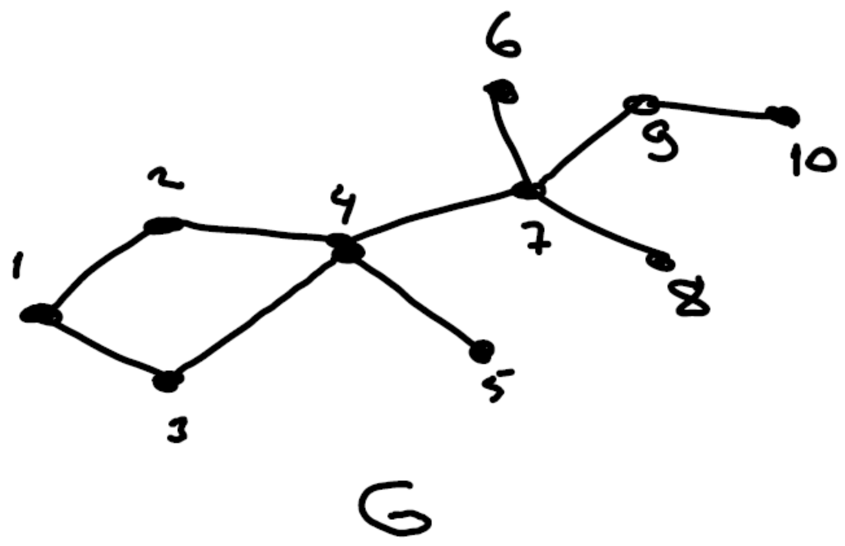
$\text{diam}(G) = 5$ (Çap)

$r(G) = 3$ (Yarı çap)

Merkez tepeler;

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

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



Kıyı tepeler;

$p(G) = \{v \mid e(v) = \text{diam}(G)\}$

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

Dijsktra Algoritması

---Dijkstra algoritması, bir tepeden diğer tüm tepelere en kısa yolları hesaplar.

---Ağırlıklı ve yönlü graflar için geliştirilmiştir. Fakat ağırlıksız ve yönsüz graflar içinde kullanılabilir.

---Dijkstra algoritması en kısa yolları Greedy yaklaşımı ile belirler.

---Yani bir tepeden diğer bir tepeye geçerken olası en iyi yerel çözümü göz önüne alır.

---Her seferinde bir sonraki tepeye ilerleme Greedy yaklaşımına göre yapılır.

Algoritma:

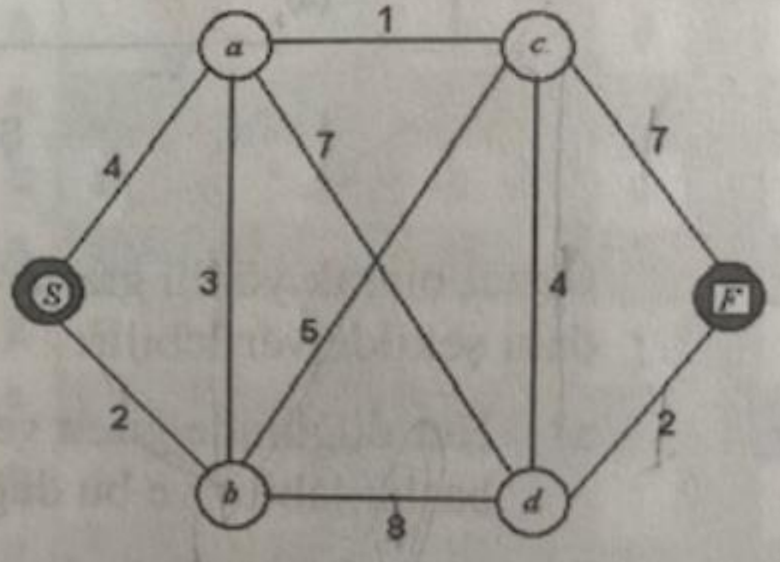
```
Dijkstra_uzaklık (graf G, düğüm v0)
// v0 başlangıç düğüm
{
    S = {v0};
    for(i=1; i<n; i++)
            D[i]=C[v0,i];          // C[i,j] - maliyet matrisi
    for (i=0; i<n-1; i++)
         {
    V-S'den öyle bir w düğümü seçelim ki D[w] minimum olsun;
            w'yi S'ye ekle;

    for v∈V - S
    D[v]=min(D[v],D[w]+C[w,v]);     // uzaklıkları yeniden ayarla
            }
}
```

C kodu çalışma olarak bırakılmıştır!!!

Örnek: Örnek, 5 numaralı kaynaktan alınmıştır.

S den F ye en kısa yolu bulunuz.

$$
A=\begin{array}{c} \\ S \\ a \\ b \\ c \\ d \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} S & a & b & c & d & F \end{array} \\
\left|\begin{array}{cccccc}
0 & 4 & 2 & \infty & \infty & \infty \\
4 & 0 & 3 & 1 & 7 & \infty \\
2 & 3 & 0 & 5 & 8 & \infty \\
\infty & 1 & 5 & 0 & 4 & 7 \\
\infty & 7 & 8 & 4 & 0 & 2 \\
\infty & \infty & \infty & 7 & 2 & 0
\end{array}\right|
\end{array}
$$



S den uzaklıklar…

Tepeler	S	a	b	c	d	F
Uzaklık	0	4	2	∞	∞	∞

1 . adım: a,b,c,d,F arasından minimum olan b=2

Tepeler	S	a	b*	c	d	F
Uzaklık	0	4	2	∞	∞	∞

(b,a) için $D(a) = \min(4, 2+3) = 4$

(b,c) için $D(c) = \min(\infty, 2+5) = 7$

(b,d) için $D(d) = \min(\infty, 2+8) = 10$

(b,F) için $D(F) = \min(\infty, \infty) = \infty$

Böylece; yeni tablo aşağıdaki gibidir.

Tepeler	s	a	b*	c	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞

2. adım: a,c,d,F arasından minimum olan a=4

Tepeler	s	a *	b *	c	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞

$\min\{a,c,d,F\} = 4$

(a,c) için $D(c) = \min(7, 4+1) = 5$

(a,d) için $D(d) = \min(10, 4+7) = 10$

(a,F) için $D(F) = \min(\infty, \infty) = \infty$

2. Adım sonrası yeni tablo aşağıdaki gibidir.

Tepeler	s	a	b	c	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞

3. adım: c,d,F arasından minimum olan c=5

Tepeler	s	* a	* b	* c	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞

$$\min\{c, d, F\} = 5$$

(c, d) için $D(d) = \min(10, 5+4) = 9$

(c, F) için $D(F) = \min(\infty, 5+7) = 12$

3. Adım sonrası yeni tablo aşağıdaki gibidir.

Tepeler	s	a *	b *	c *	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞
c	0	4	2	5	9	12

4. adım: d,F arasından minimum olan d=9

Tepeler	s	a*	b*	c*	d*	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞
c	0	4	2	5	9	12

$\min(d,F) = 9$

(d,F) için $D(F) = \min(12, 9+2) = 11$

4. Adım sonrası yeni tablo aşağıdaki gibidir.

		*	*	*	*	
Tepeler	s	a	b	c	d	F
s	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞
c	0	4	2	5	9	12
d	0	4	2	5	9	11

5. adım: F son tepe F=11 için tablo aşağıdaki gibidir.

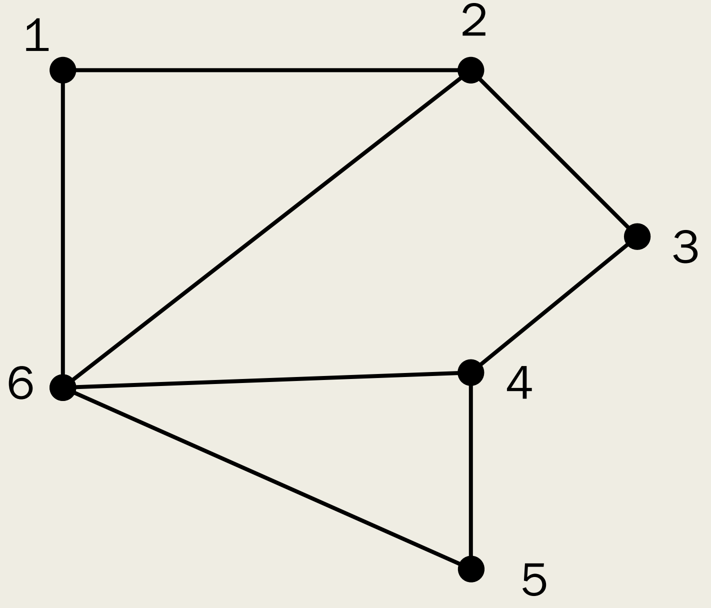
Tepeler	S	a*	b*	c*	d*	F
S	0	4	2	∞	∞	∞
b	0	4	2	7	10	∞
a	0	4	2	5	10	∞
c	0	4	2	5	9	12
d	0	4	2	5	9	11
F	0	4	2	5	9	$\boxed{11}$

S den diğer tüm tepelere en kısa uzaklıklar

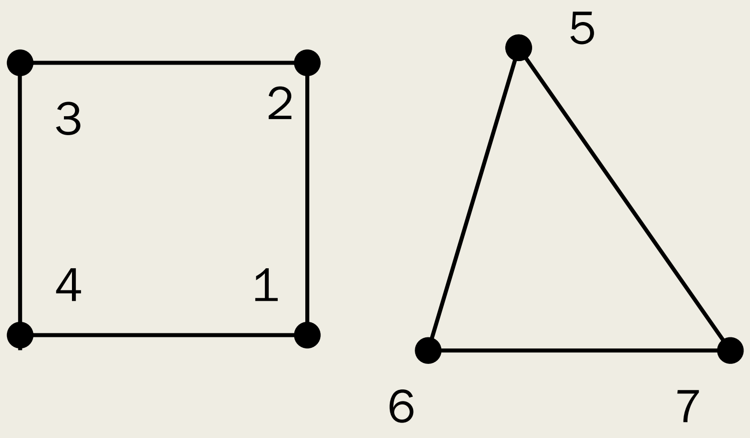
Tepeler	S'den en kısa yollar	Min. uzaklık
a	$s \to a$	4
b	$s \to b$	2
c	$s \to a \to c$	5
d	$s \to a \to c \to d$	9
F	$s \to a \to c \to d \to F$	11

Bağlantılılık (Connectivity)

Birleştirilmiş Graf: Bir G grafında herhangi iki tepe arasında en az bir tane yol varsa, grafın herhangi bir tepesinden diğer tüm tepelere gidilebiliyorsa bu grafa birleştirilmiş graf denir.



Birleştirilmiş Graf



Birleştirilmemiş Graf

Birleştirilmemiş bir graftaki her bir parçaya grafın bileşeni adı verilir(component).

Bir Graf Nasıl Bağlantılıdır?

Bir bilgisayar ağının bir graf ile temsil edildiğini farz edelim. Bu grafın bağlantılı olduğunun bilinmesi bize bu ağda iki bilgisayarın iletişim halinde olduğunu söyler. Fakat biz ağın nasıl dayanıklı olduğunu anlamak isteriz. Örneğin, bir iletişim hattı arızalandığında veya yönlendiricilerden sonra bütün bilgisayar iletişimi için bu hala mümkün olabilecek mi? Bunu ve benzer soruları cevaplamak için, bazı kavramlar geliştirilmiştir.

Bazen bir tepe ve bitişik tüm ayrıtlar graftan kaldırıldığında birden fazla bağlantılı bileşen elde edilir. Bu tepeler kesim tepeler (veya birleşim noktaları) olarak çağırılır. Bir kesim tepe graftan kaldırıldığında bağlantılı olmayan bir alt graf üretilir. Benzer olarak, kesim ayrıt veya köprü olarak adlandırılan bir ayrıt graftan kaldırıldığında orijinal graftan daha fazla bileşene sahip bağlantılı bir graf oluşur. Unutmayalım ki, bir graf ile temsil edilmiş bilgisayar ağında, bir kesim tepe ve bir kesim ayrıt, gerekli yönlendiriciyi ve zarar görmemiş gerekli bir iletişim hattını temsil eder.

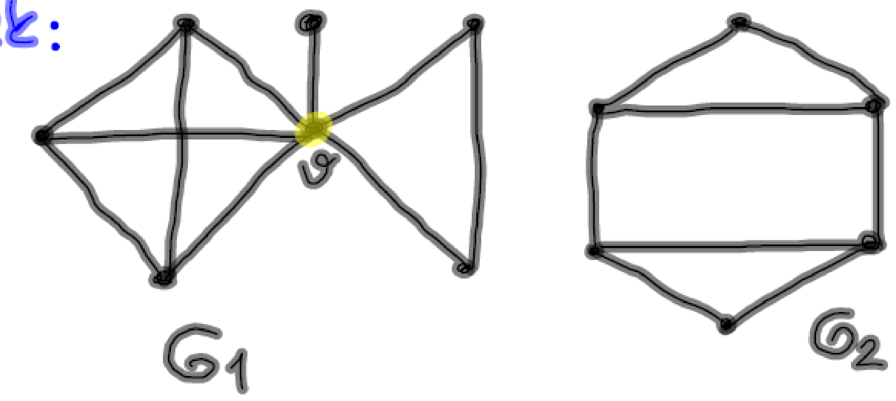
Tanım:Eğer G grafında (u,v)-yolu mevcut ise u,v ayrık tepeleri G grafında bağlıdır.Eğer tüm tepe çiftleri bağlıysa G grafı da bağlıdır.

Tanım:Eğer H grafı, bağlantılı ve G nin daha fazla tepe ya da ayrıt içeren bağantılı bir alt grafı tarafından içerilmeyen bir graf ise G grafının H alt grafına G nin bir bileşeni denir ve G nin bileşenlerinin sayısı w(G) ile gösterilir.

Tanım: Bir G grafı için $V^* \subset V(G)$ ve $E^* \subset E(G)$ olsun.Eğer $w(G-V^*)>w(G)$ ise V^* bir '*vertex cut*' olarak adlandırılır.Eğer V^* yalnızca bir tepeden oluşuyorsa, v bir '*cut vertex*' olarak adlandırılır.Benzer şekilde, eğer $w(G-E^*)>w(G)$ ise E^* bir '*edge cut*' olarak adlandırılır.Eğer E^* yalnızca bir ayrıttan oluşuyorsa, e '*cut edge*' olarak adlandırılır.

Tanım: Bir G grafının bir v tepesi için $w(G-v) > w(G)$ oluyor ise v'ye kesim tepe denir.

Örnek:



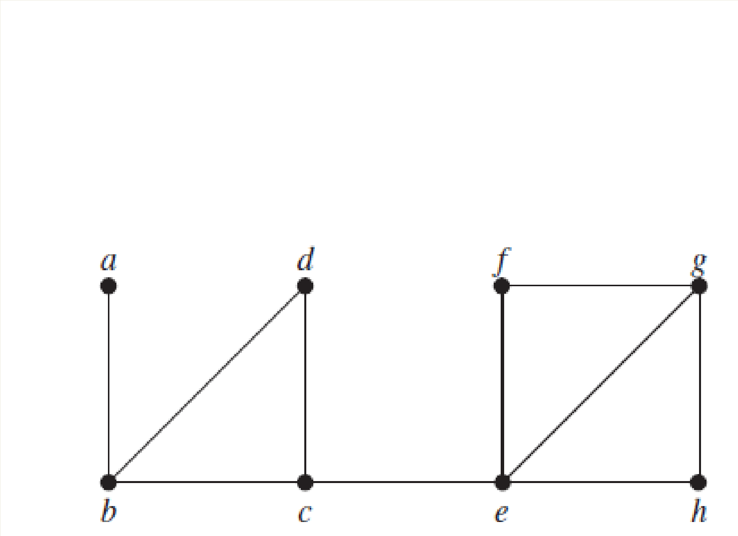
$w(G_1) = 1$

$w(G_1 - v) = 3$

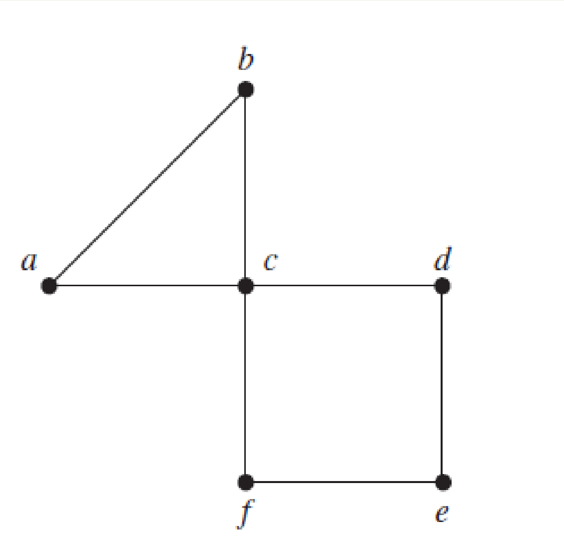
G_1 grafında v bir kesim tepedir.

G_2 grafında bir kesim tepe yoktur.

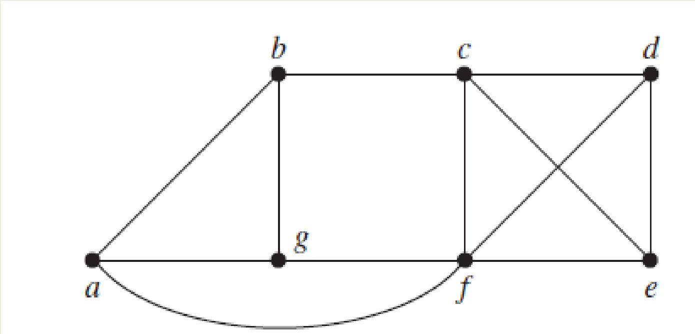
--- Aşağıdaki grafları bağlantısız yapmak için graftan atılması gerekli tepeler hangileridir?



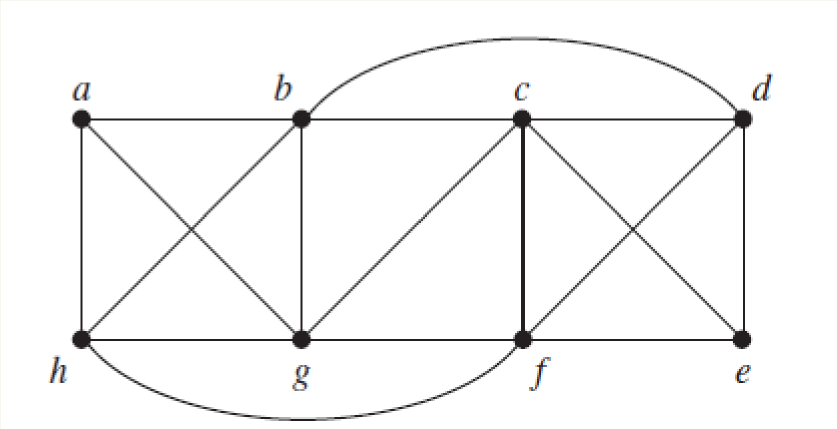
1



2

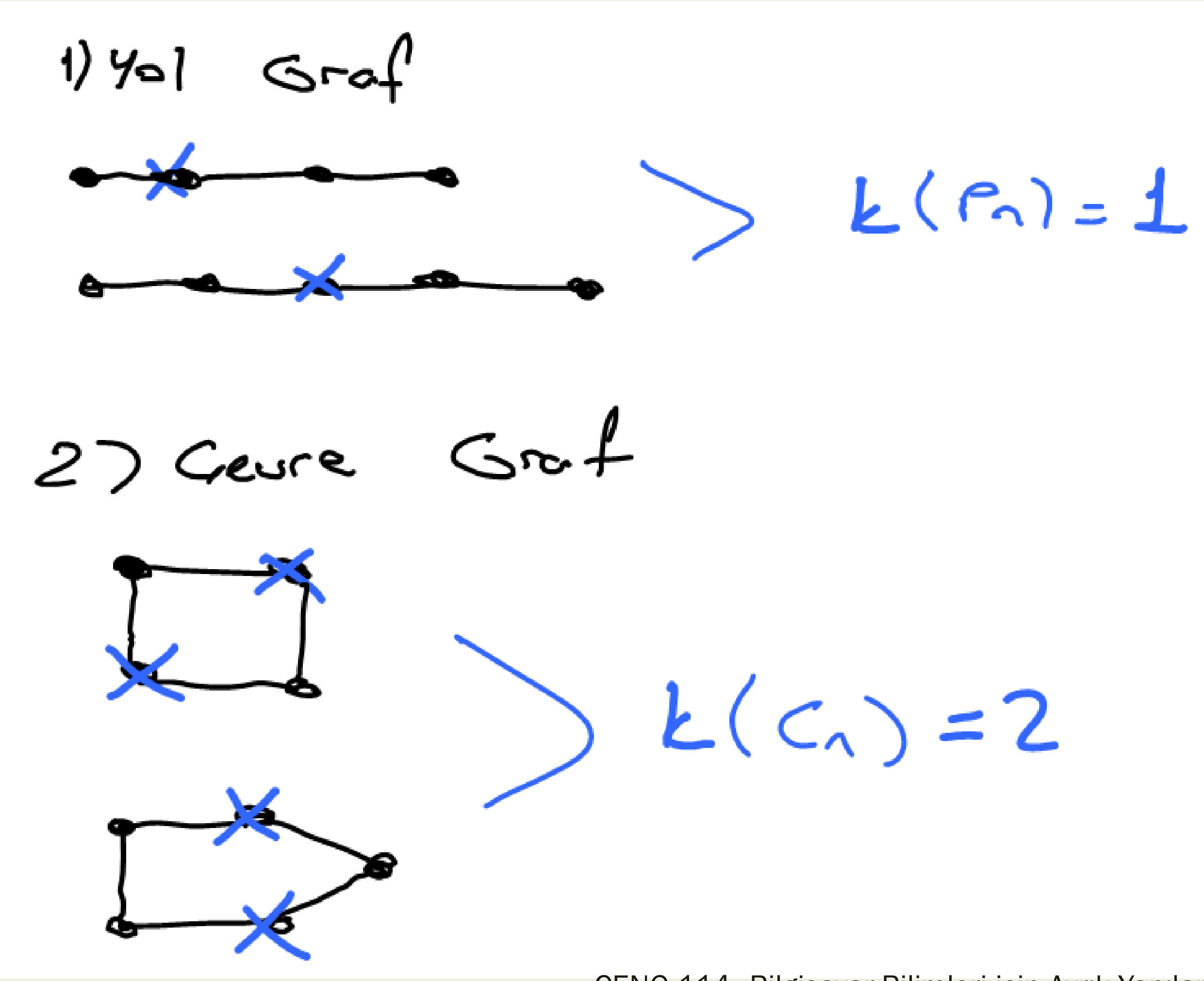


3

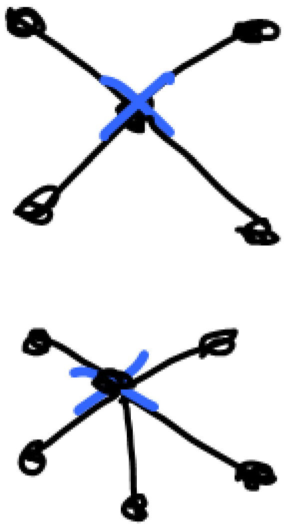


4

Tanım: Bağlantılı bir *G* grafını bağlantısız yapmak veya tek izole tepe elde etmek için graftan çıkarılması gereken en az tepe sayısına grafın **bağlantılılığı (connectivity)** denir ve *k(G)* ile gösterilir. Önemli grafların connectivity değerleri aşağıdadır.

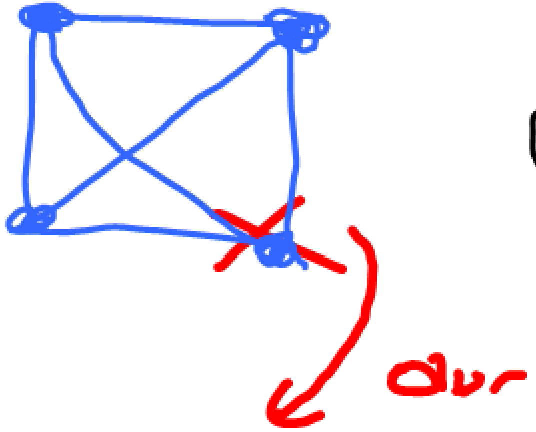


3) Yıldız Graf



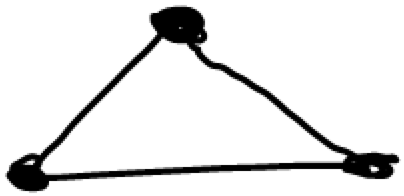
$$k(K_{1,n}) = 1$$

4) Tam Graf



K_4

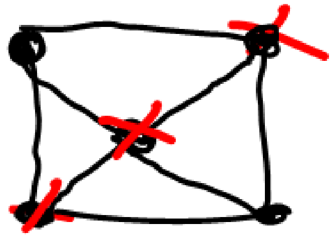
$\kappa(K_n) = n-1$



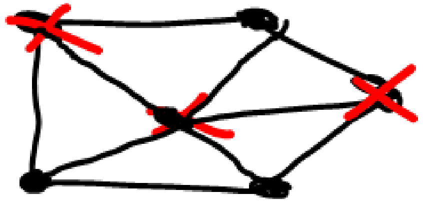
K_3

izole tepe kalmayana kadar tepe atılır.

5) Tekerlek Graf



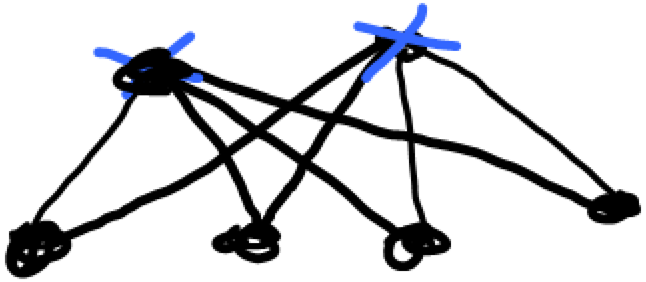
$\omega_{1,4}$



$\omega_{1,5}$

$\Big\} \; k(\omega_{1,n}) = 3$

6) iki parçalı tam graf



$\Rightarrow \; k(K_{m,n}) = \min\{n,m\}$

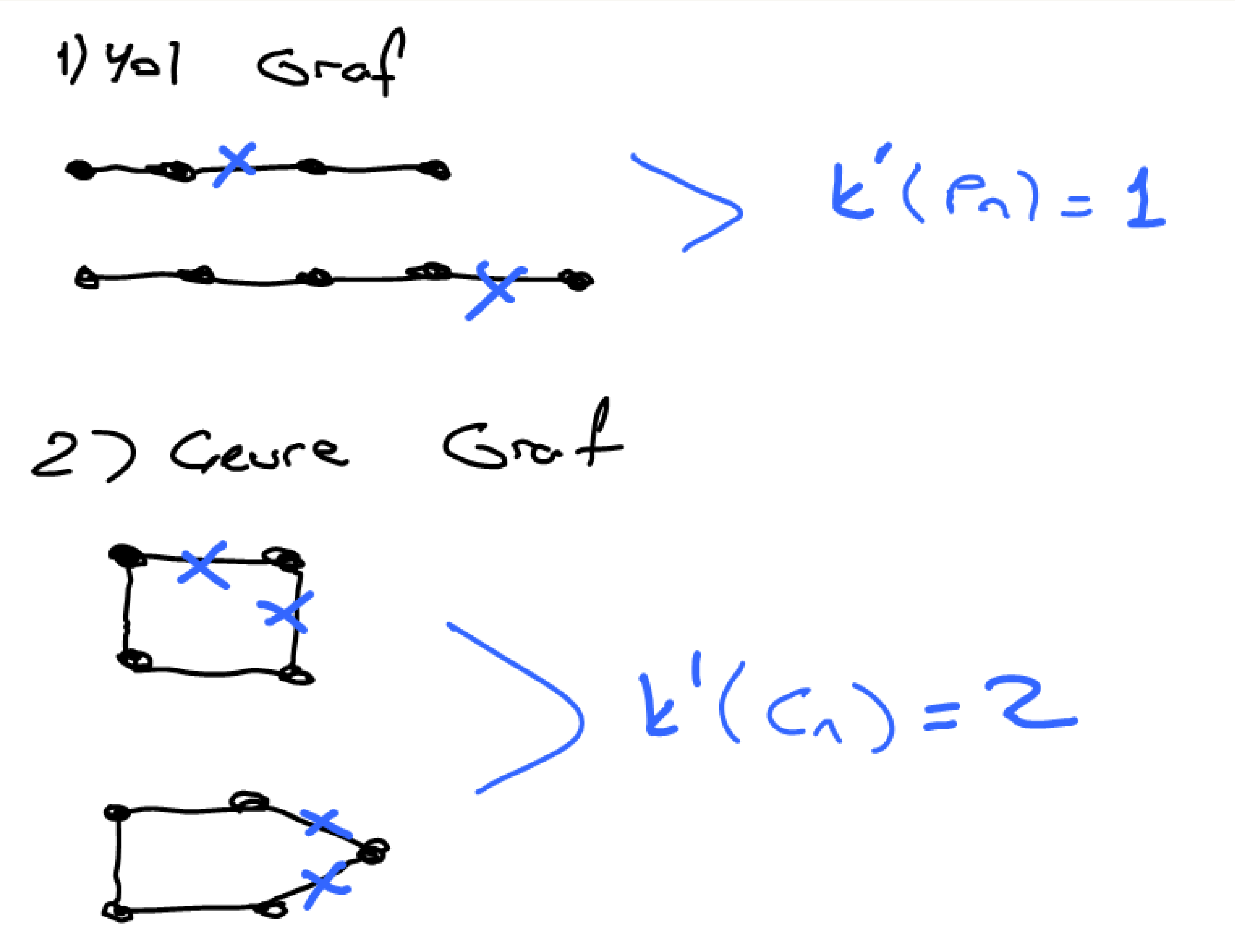
Tanım: Bağlantılı bir G grafında eğer k(G)=t ise, bu grafa t- bağlı graftır denir.

Örnek 1: n-tepeli tam graf, (n-1)- bağlı graftır.

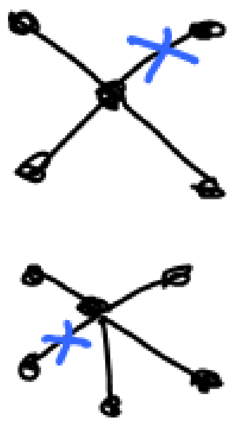
Örnek 2: n-tepeli çevre graf, *2*- bağlı graftır.

Tanım: Bir G grafını bağlantısız yapmak için graftan çıkarılması gereken en az ayrıt sayısına ayrıt bağlantılılık sayısı (edge connectivity number) denir ve $k'(G)$ ile gösterilir.

Önemli grafların ayrıt connectivity değerleri:

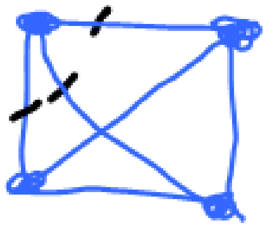


3) Yıldız Graf



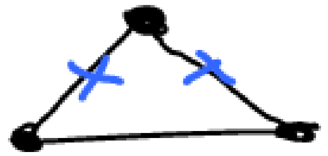
$k'(K_{1,n}) = 1$

4) Tam Graf



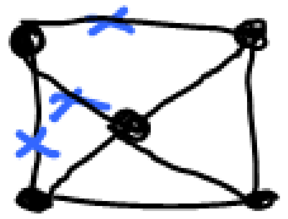
K_4

$k'(K_n) = n-1$

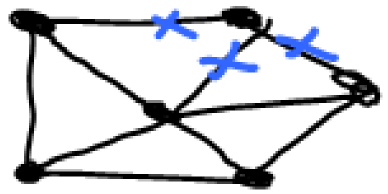


K_3

5) Tekerlek Graf



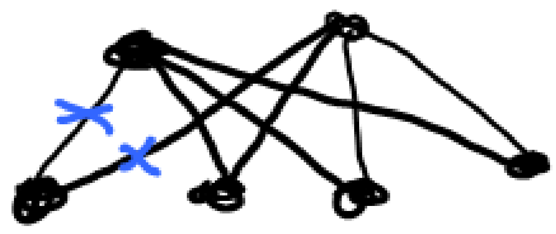
$W_{1,4}$



$W_{1,5}$

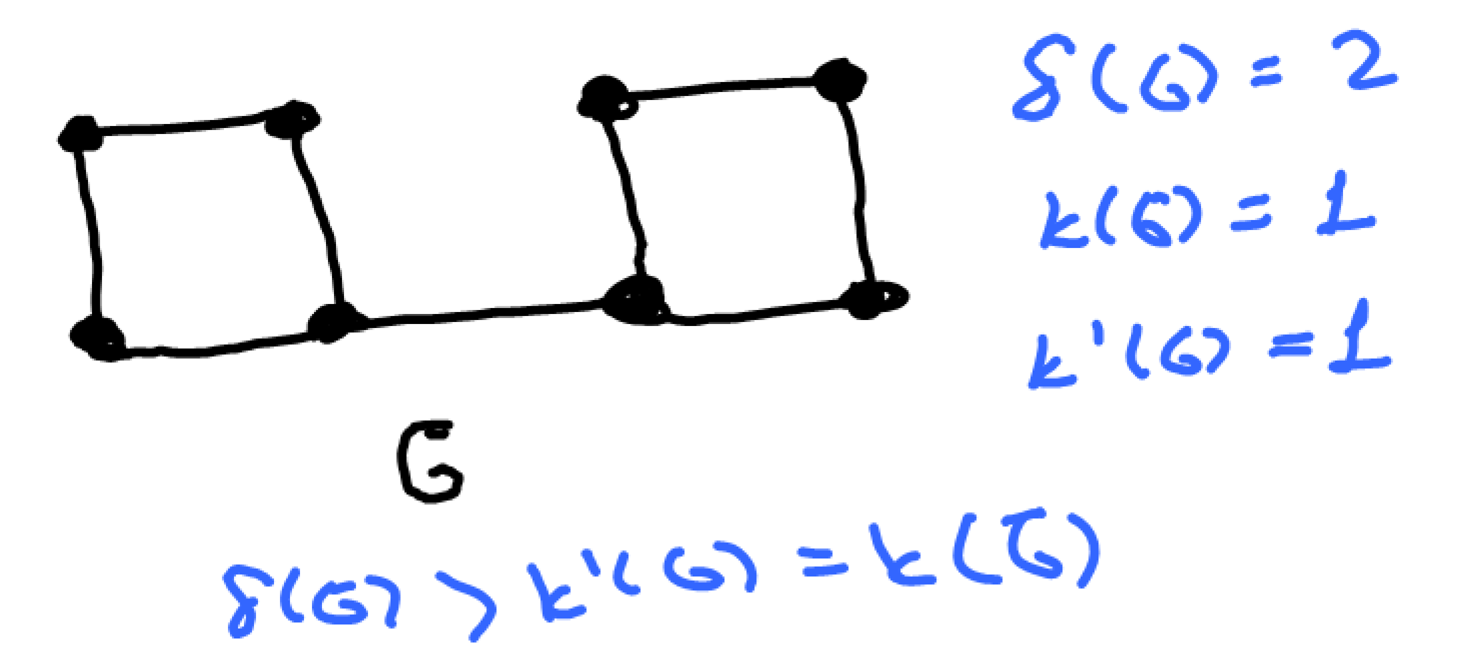
$\kappa'(W_{1,n}) = 3$

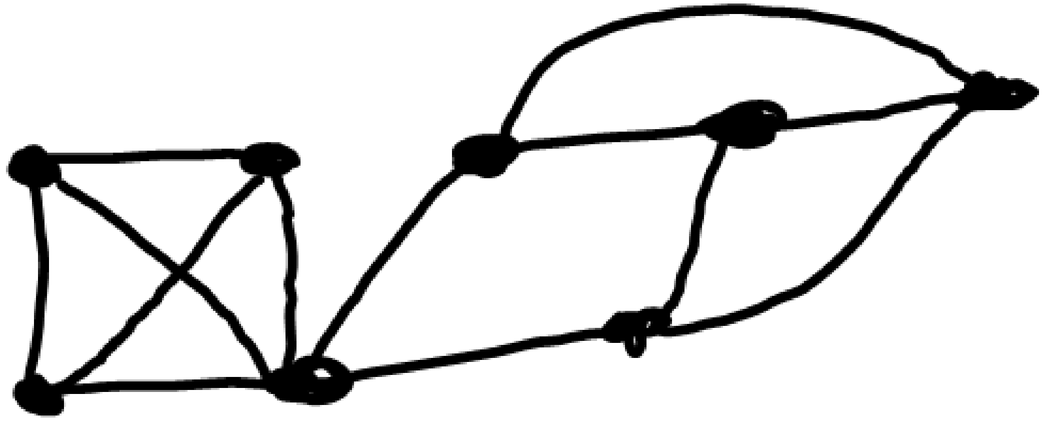
6) iki parçalı tam graf



$\Rightarrow \quad \kappa(K_{m,n}) = \min\{n,m\}$

Teorem : n tepeli herhangi bir G grafı için, $\delta(G) \geq k'(G) \geq k(G)$ dır.





G

$\delta(G) = 3$

$\kappa(G) = 1$

$\kappa'(G) = 2$

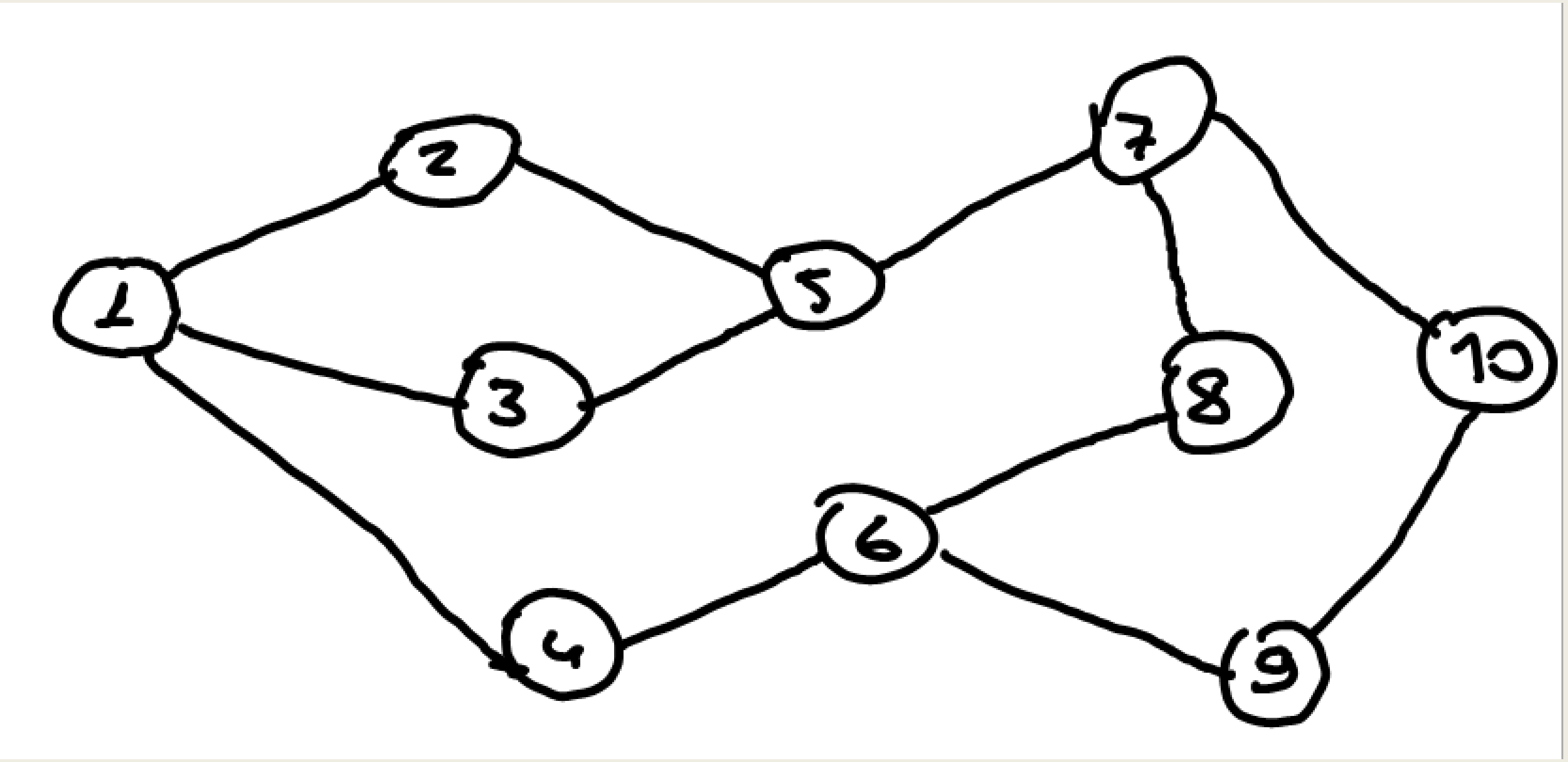
$\delta(G) > \kappa'(G) > \kappa(G)$

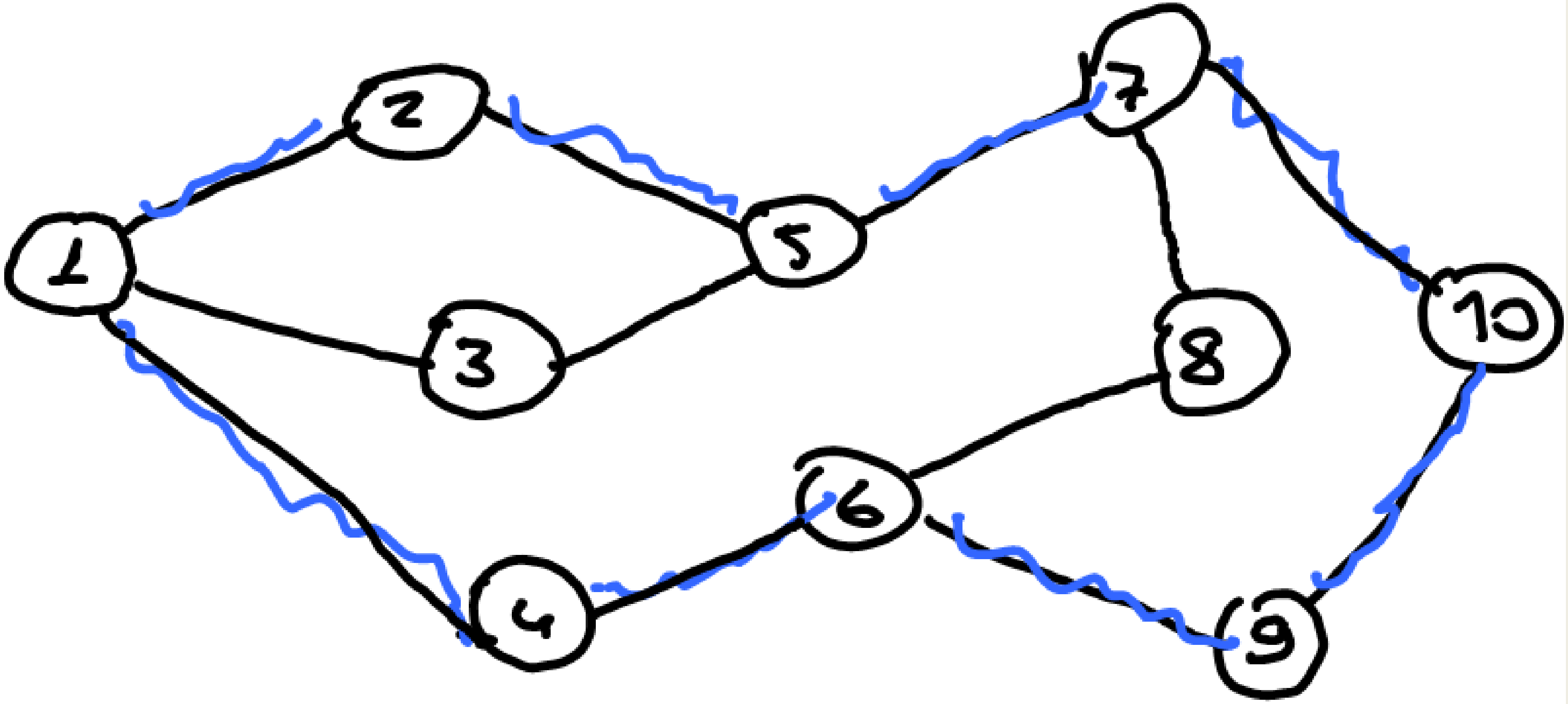
Bir Connectivity Değerinin Bulunması

Ağlardaki maksimum akış minimum kesme ye dayanan Menger'in teoremi birleştirilmişlik konusundaki en temel teoremdir ve bu teoreme göre bir grafta bitişik olmayan *u* ve *v* tepeleri arasındaki tepe tekrarsız yolların maksimum sayısı *u* ve *v* yi bağlantısız yapmak için çizgeden atılması gereken minimum tepe sayısına eşittir. Bu haliyle teorem grafın bağlantısız olmasını sağlayan tepelerin hangileri olduğu ile ilgili bir bilgi vermemektedir.

Teorem . *Bağlantılı bir G grafında ayrık ve komşu olmayan iki tepe u ve v olsun. Buna göre, G'deki içten ayrık u – v yollarının maksimum sayısı, u'yu v'den ayırmak için atılması gerekli tepelerin minimum sayısına eşittir.*

ÖRNEK: Aşağıda 10 tepeli bir G grafı verilmiştir. İçten ayrık yolların sayısı nedir?





1,2,5,7,10

1,4,6,9,10

yolları 5 ve 6 tepelerini kullanan içten ayrık yollardır.

5 ve 6 tepelerini kullanan farklı

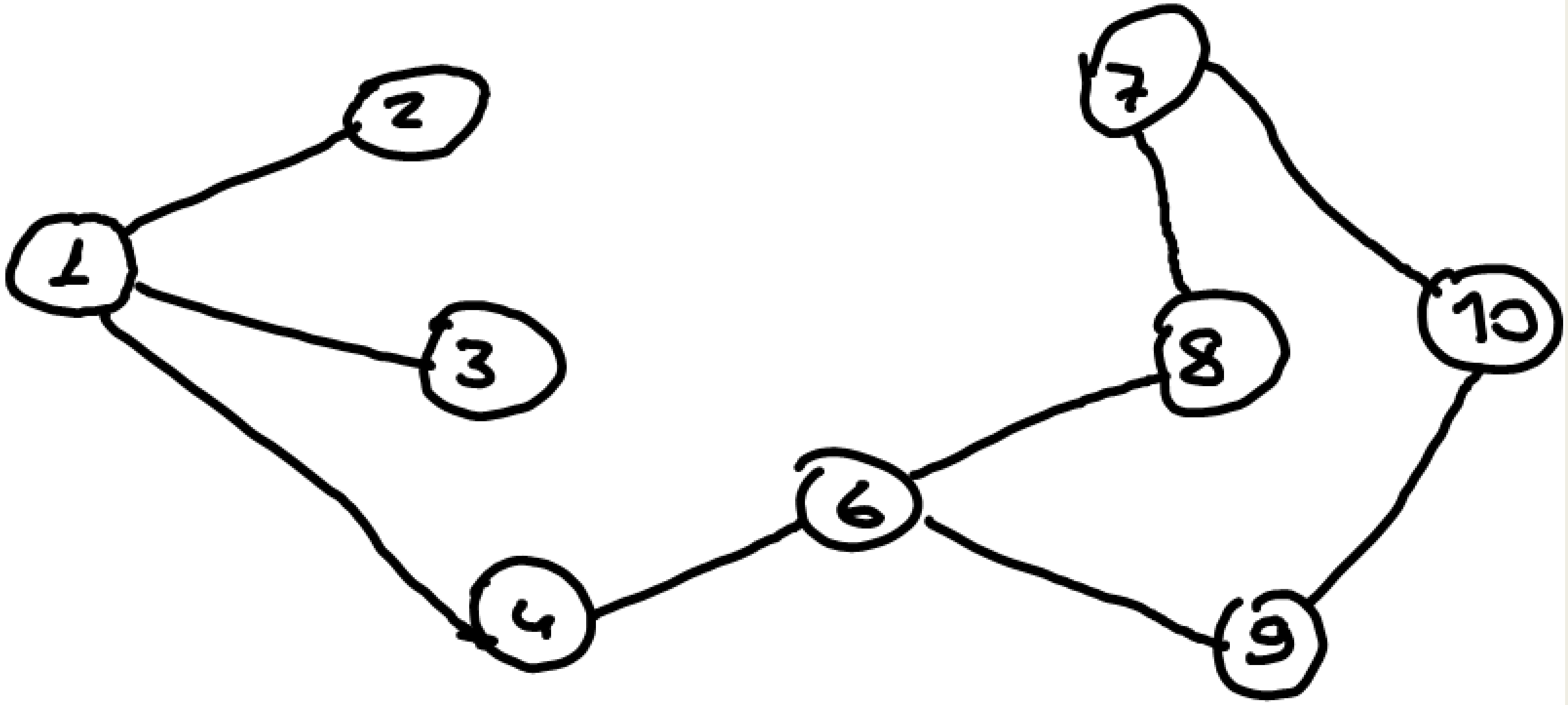
içten ayrık yollarda bulunabilir.

Fakat bunların sayısı maksimum

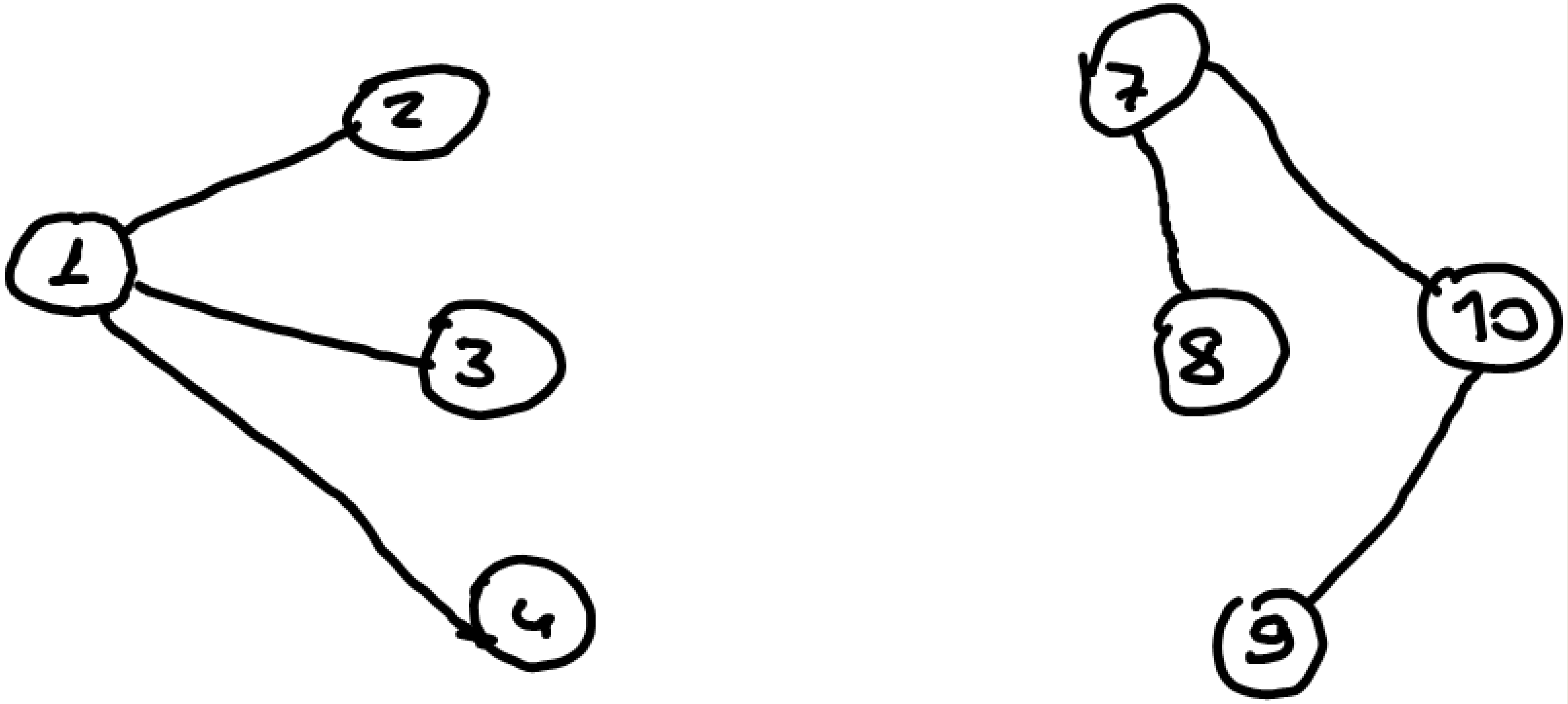
2'dir.

Böylece $k(G) = 2$ elde edilir.

5 tepesi silindiğinde



5 ve 6 tepeleri silindiğinde:



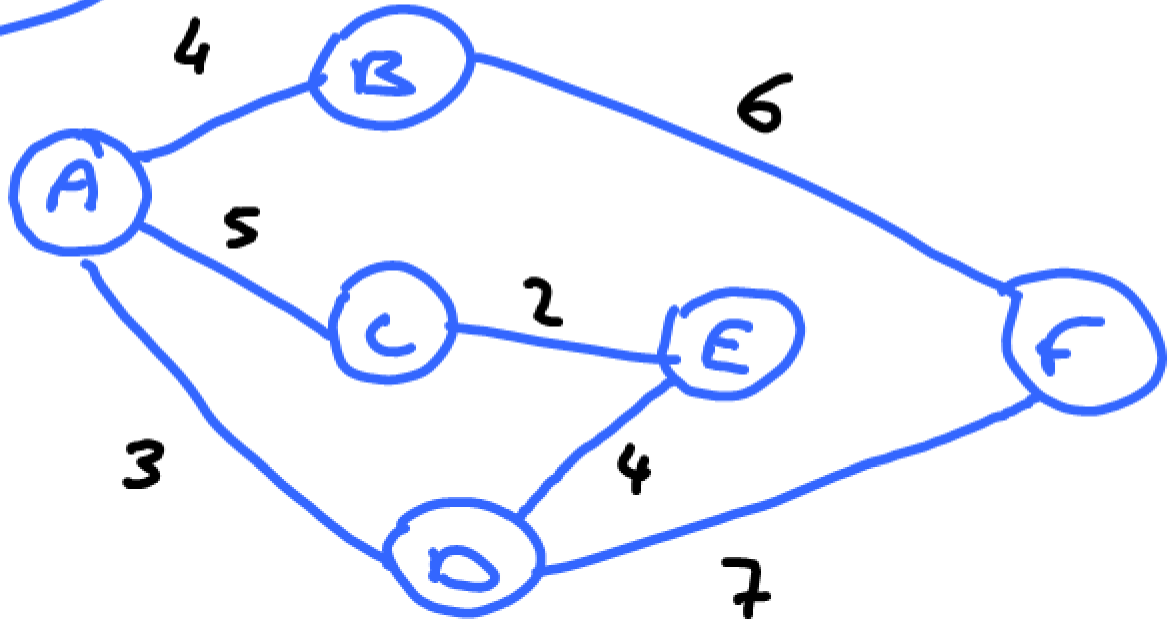
* içten ayrık yolların sayısı Ford-Fulkerson (en büyük akış) algoritması yardımıyla bulunabilir.

Ford-Fulkerson Sözde Kodu

ford-fulkerson (G, s, t)
for each edge $(u,v) \in E(G)$
$f[u,v] = 0$
$f[v,u] = 0$
end

while (s'den t'ye p yolu var ise)
$c_f(p) = \min\{ c_f(u,v) \mid (u,v) \in p \}$
for each edge (u,v) in p
$f(u,v) = f(u,v) + c_f(p)$
$f(v,u) = -f(u,v)$
end
end

Örnek

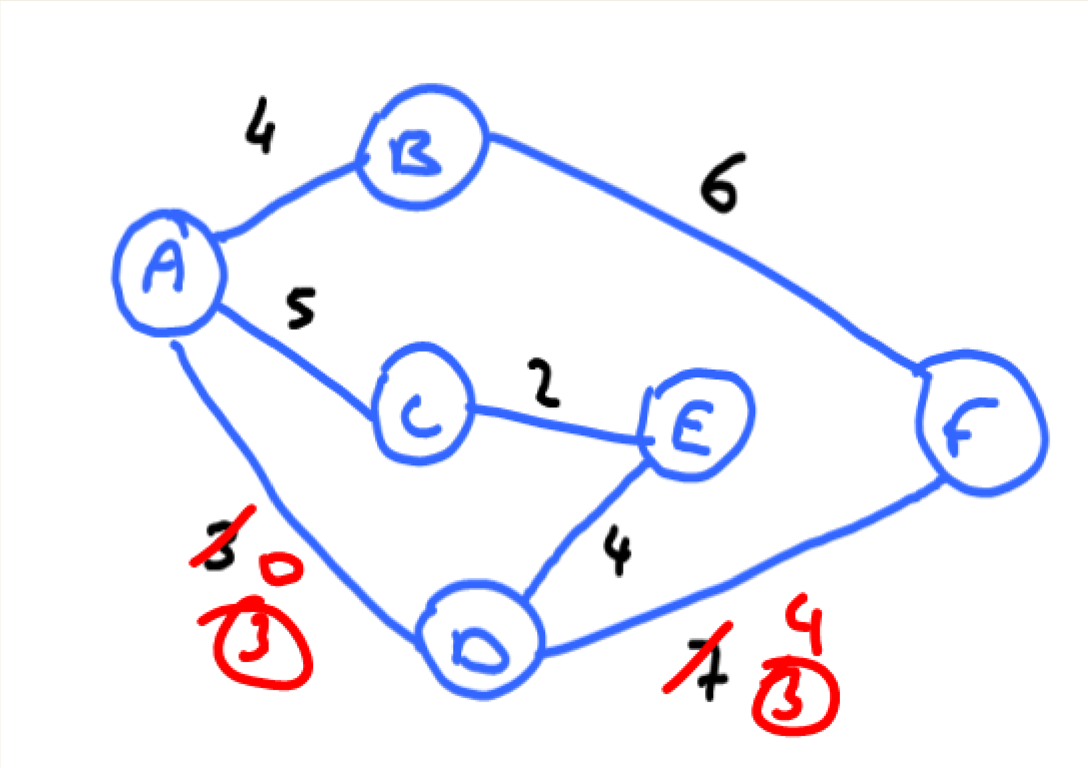


A→F'ye en büyük akış?

1. adım: A→F ye yollar bulunur.

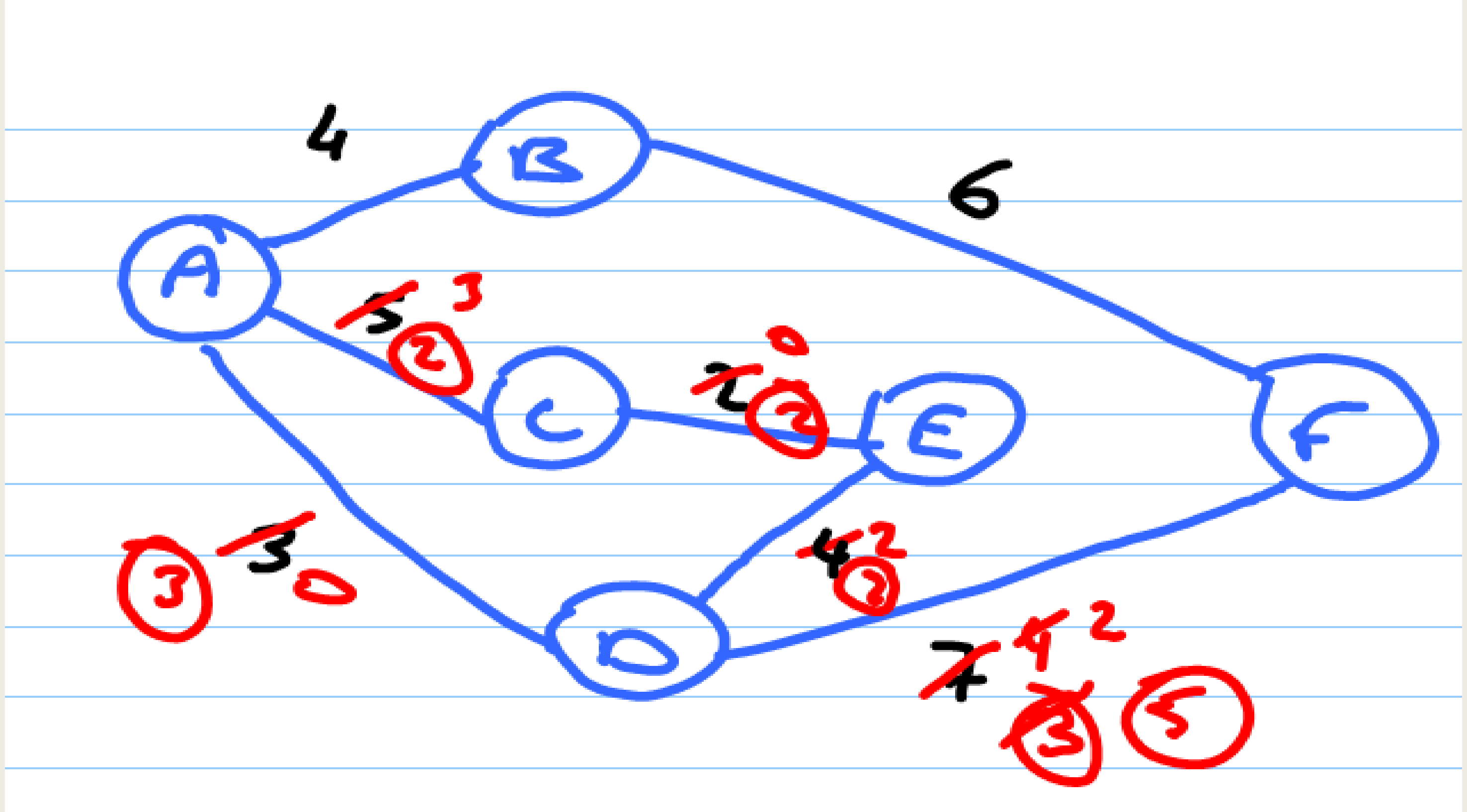
ADF yolu

$$\min\{\underset{3}{A-D}, \underset{7}{D-F}\} = 3 \,//$$



2. adım: ACEDF yolu

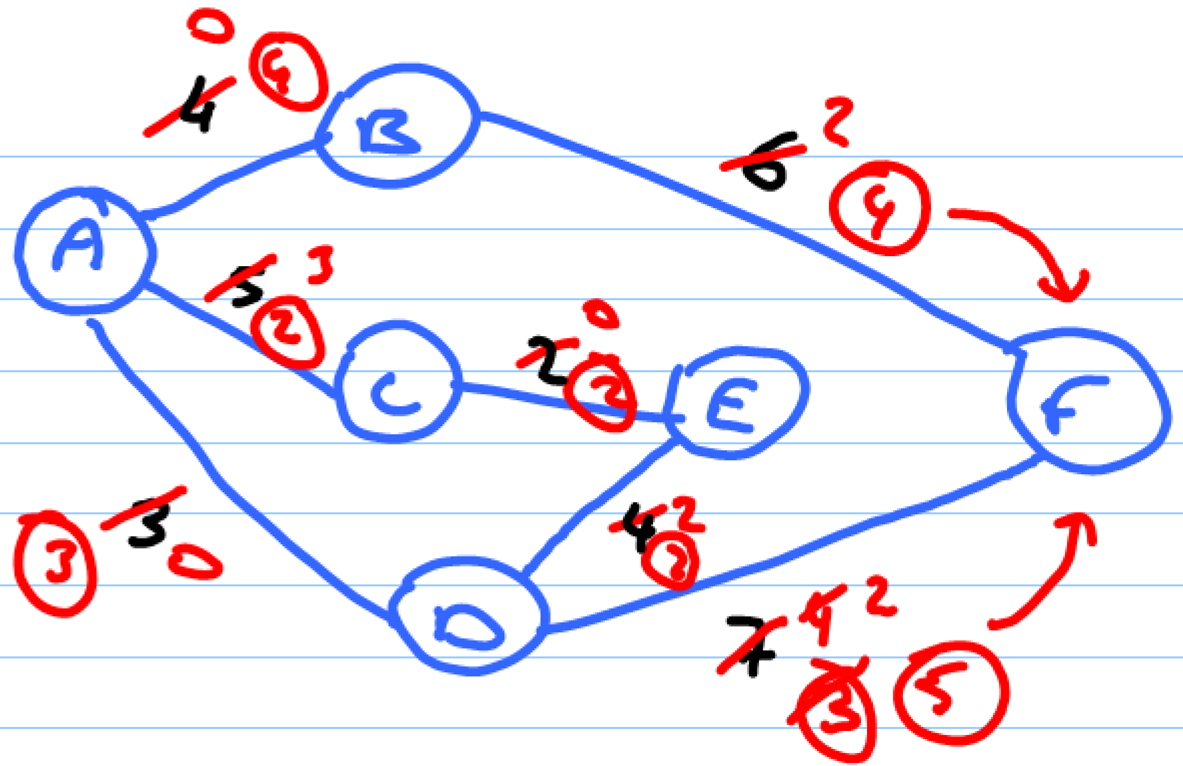
$$\min\{\underset{5}{A-C},\ \underset{2}{C-E},\ \underset{4}{E-D},\ \underset{4}{D-F}\} = 2$$



3. adım

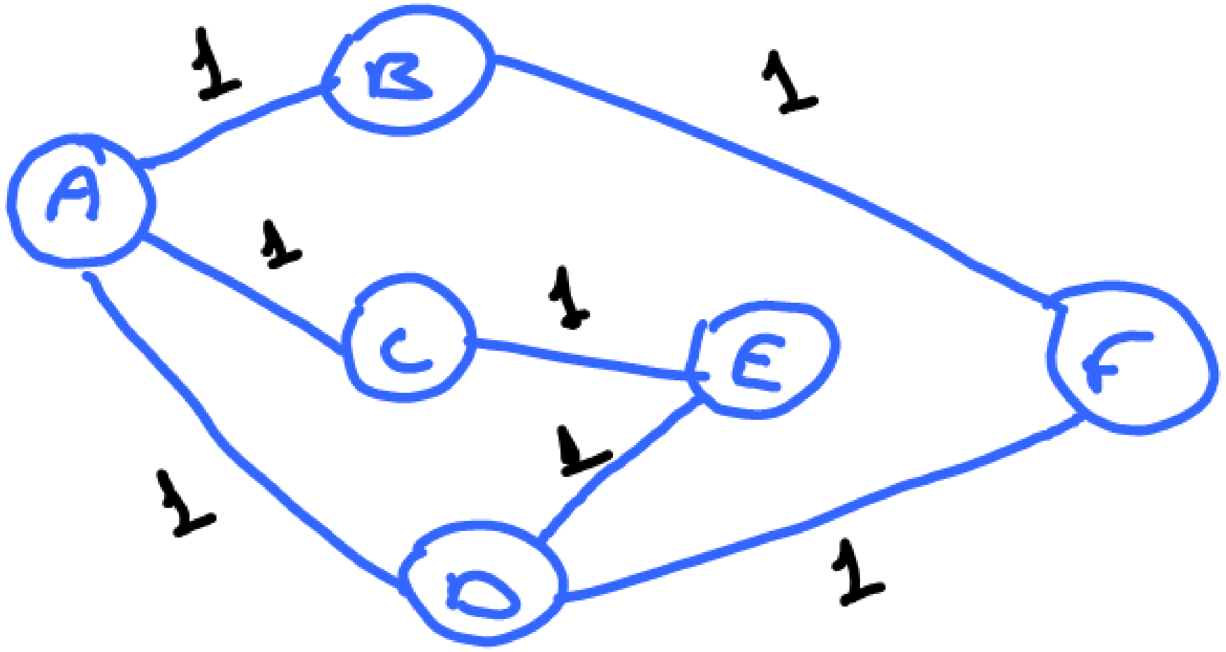
ABF yolu $\min\{A\text{–}B,\ B\text{–}F\} = 4$

$\qquad\qquad\qquad\quad\ \ 4 \qquad\quad 6$



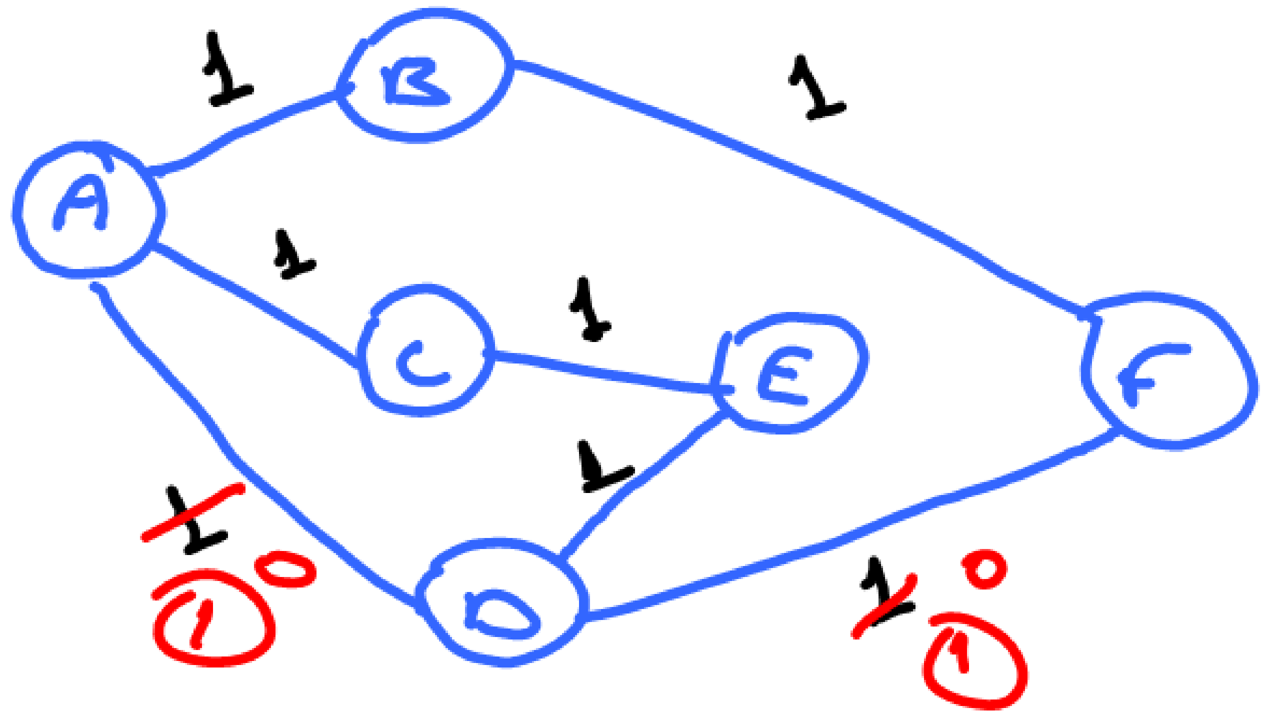
A'dan F'ye $4+5=9$ birim akış oldu.

* Tüm akış değerlerine 1 verdiğimizde, aşağıdaki graf elde edilir.



1. adım: ADF yolu

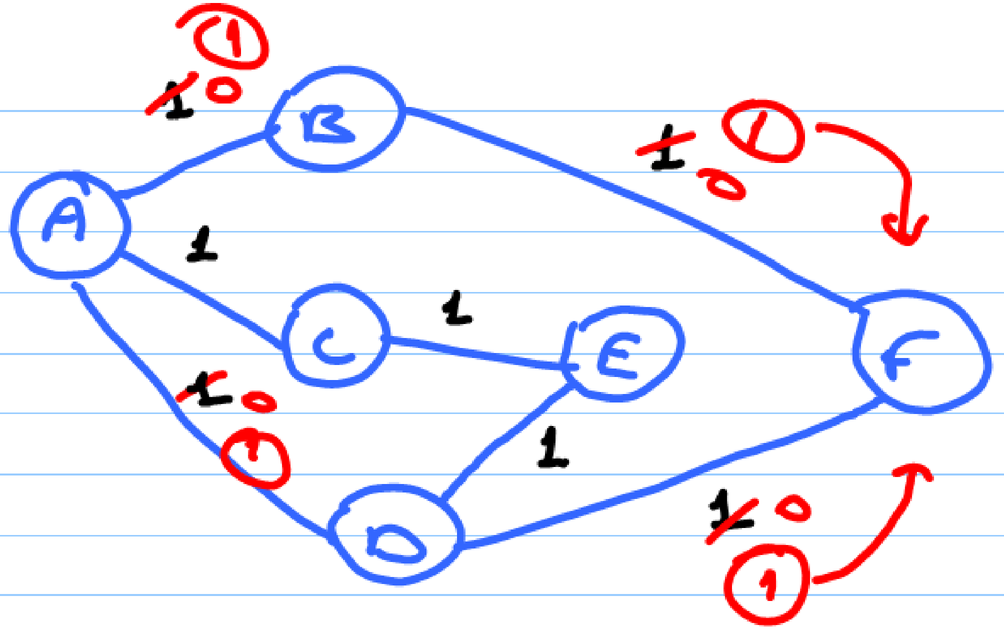
$$\min\{\underset{1}{A\text{-}D},\ \underset{1}{D\text{-}F}\} = 1$$



2. adım: ACEDF yolu

$$\min\{\underset{1}{A-C},\ \underset{1}{C-E},\ \underset{1}{E-D},\ \underset{0}{D-F}\} = 0$$

3. adım) ABF yolu $\min\{\underset{1}{A-B},\ \underset{1}{B-F}\} = 1$

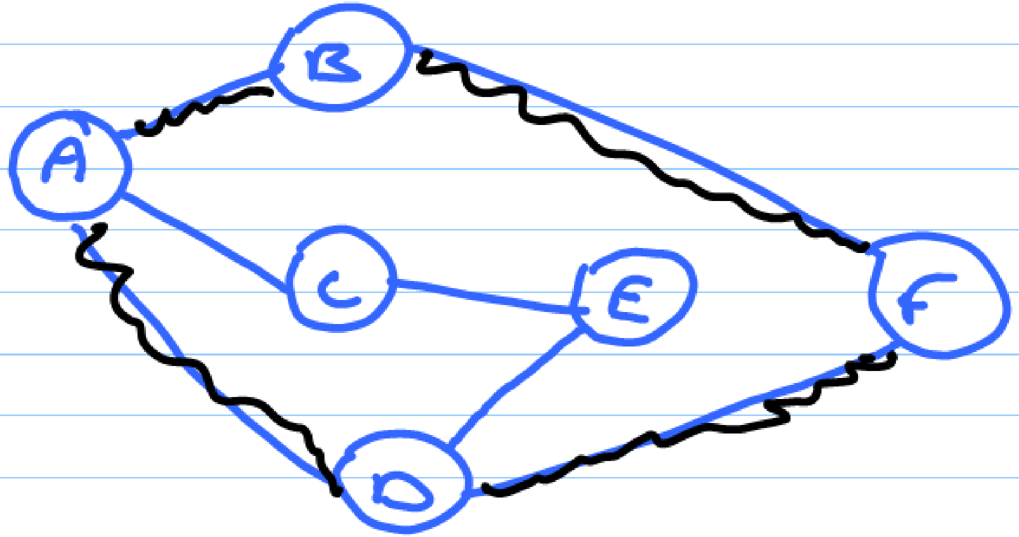


$1+1=2$

mak. akış

içten ayrık yolların sayısı

Sonuç olarak: içten ayrık yolların sayısı 2'dir.



Böylece $k(G) = 2$ 'dir.

* içten ayrık yollarda ortak ayrıt yoktur!!!

Kaynaklar

- *1- Discrete Mathematics and Its Applications*, Kennet H. Rosen

(Ayrık Matematik ve Uygulamaları, Kennet H. Rosen (Türkçe çeviri), Palme yayıncılık)

- *2- Discrete Mathematics: Elementary and Beyond*, L. Lovász, J. Pelikán, K. Vesztergombi, 2003.
- *3- Introduction to Algorithms*, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, 2009.
- *4- Introduction To Design And Analysis Of Algorithms*, A. Levitin, 2008.
- *5-Algoritmalar (Teoriden Uygulamalara)*, Vasif V. Nabiyev, Seçkin Yayıncılık