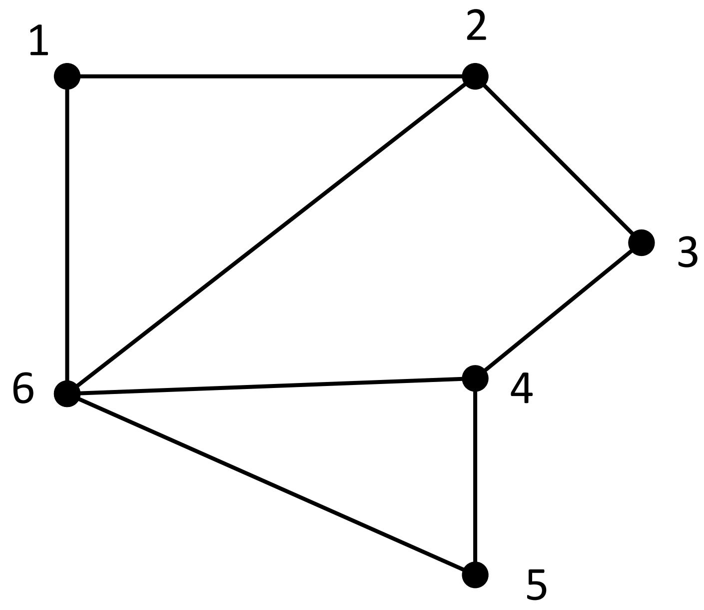
BİLGİSAYAR BİLİMLERİNDE GÜNCEL KONULAR II

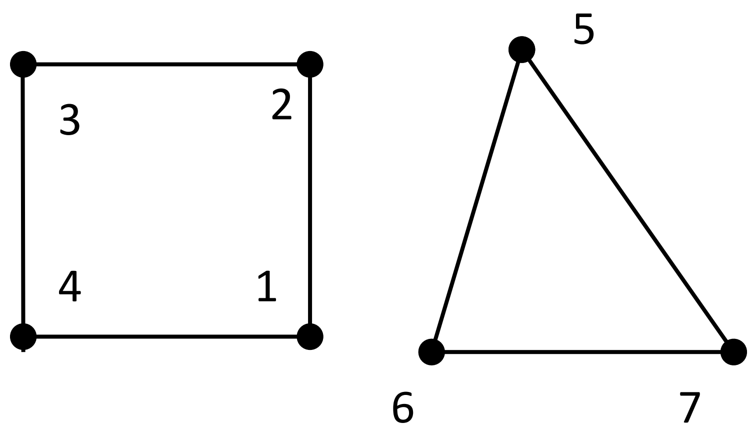
Hafta 11

. 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.

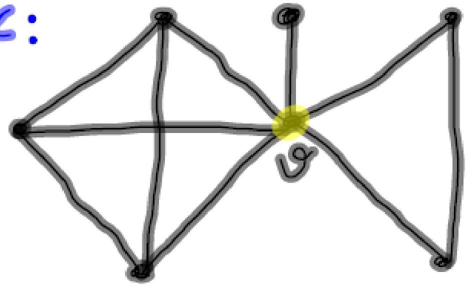
Tanım:Eğer G grafında (u,v)-yolu mevcut ise u,v ayrık tepeleri G grafında bağlıdır.Eğer tüm tepe çiftleri bağlıysa G grafi da bağlıdır.

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

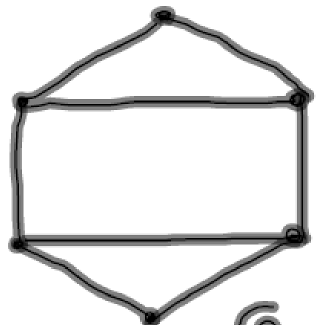
Tanım: Bir G grafı için $V^* \subset V(G)$ ve $E^* \subset E(G)$ olsun.Eğer $w(G-V^*)>w(G)$ ise V^* bir '*vertex cut*' olarak adlandırılır.Eğer V^* yalnızca bir tepeden oluşuyorsa, v bir '*cut vertex*' olarak adlandırılır.Benzer şekilde, eğer $w(G-E^*)>w(G)$ ise E^* bir '*edge cut*' olarak adlandırılır.Eğer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



G_1



G_2

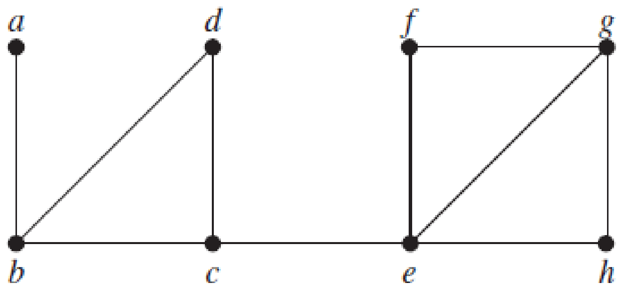
$w(G_1) = 1$

$w(G_1 - v) = 3$

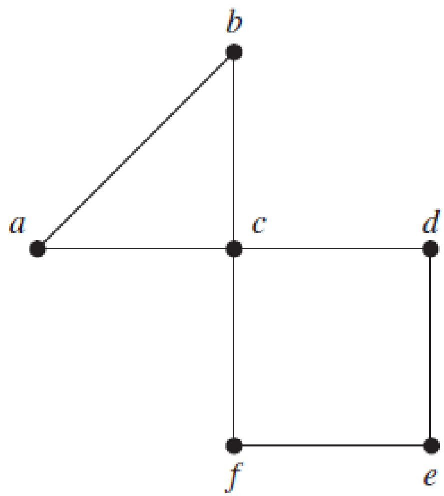
G_1 grafında v bir kesim tepedir.

G_2 grafında bir kesim tepe yoktur.

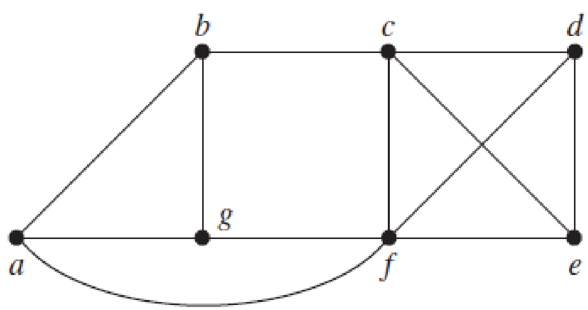
-- Aşağıdaki grafları bağlantısız yapmak için graftan atılması gerekli tepeler hangileridir?



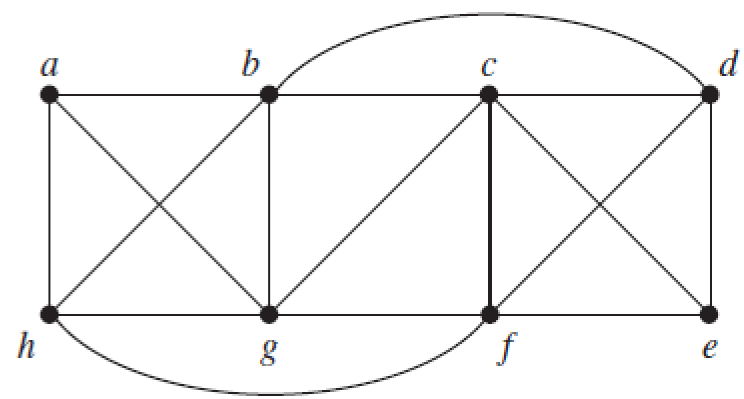
1



2



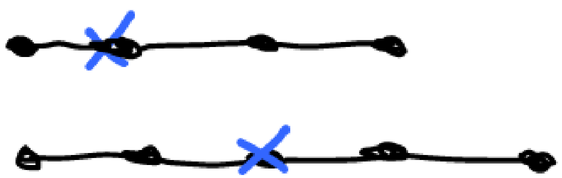
3



4

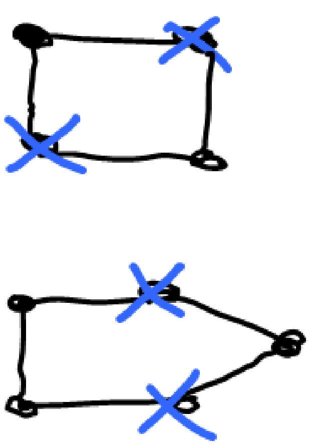
Tanım 1: Bağlantılı bir *G* grafını bağlantısız yapmak veya tek izole tepe elde etmek için graftan çıkarılması gereken en az tepe sayısına grafın **bağlantılılığı (connectivity)** denir ve *k(G)* ile gösterilir. Önemli grafların connectivity değerleri aşağıdadır.

1) Yol Graf



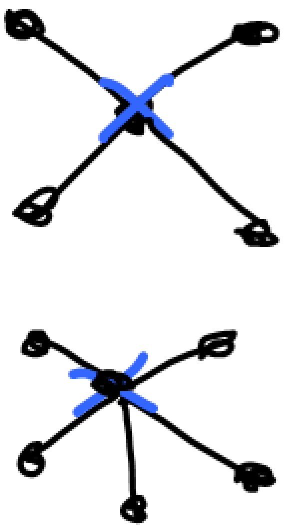
$k(P_n) = 1$

2) Çevre Graf



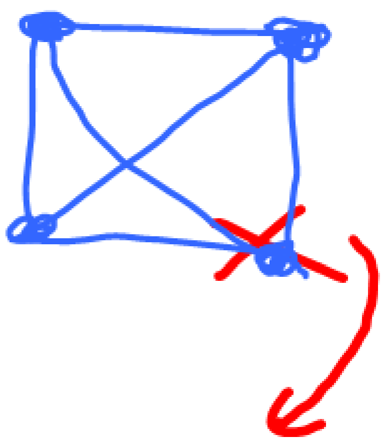
$k(C_n) = 2$

3) Yıldız Graf



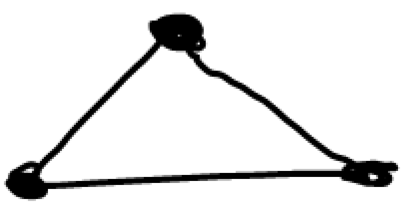
$k(K_{1,n}) = 1$

4) Tam Graf



dur

K_4

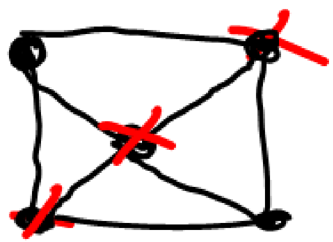


K_3

$k(K_n) = n-1$

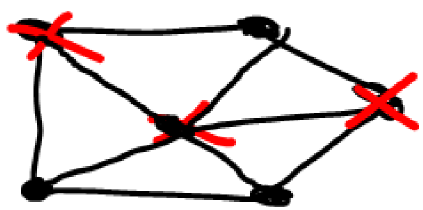
izole tepe kalmayana kadar tepe atılır.

5) Tekerlek Graf



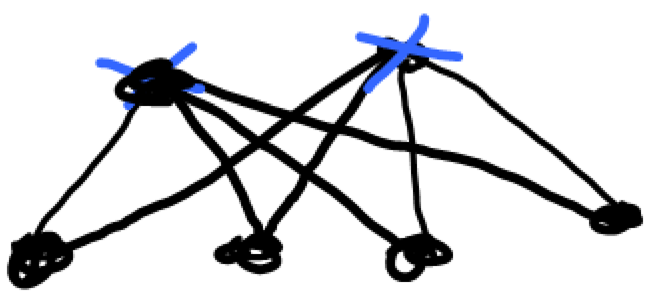
$W_{1,4}$

$\Bigg\} k(W_{1,n}) = 3$



$W_{1,5}$

6) iki parçalı tam graf



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

Tanım: G grafı eğer $\kappa(G)=n$ n-bağlı graftır denir.

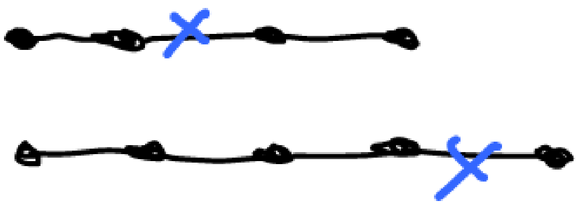
Teorem: G, 2-bağlı bir graf ve u ve v bu grafın 2 tepesi olsun. Bu durumda G'de bir C çevrimi vardır öyleki u ve v bu döngüde yer alırlar.

- K_n den bir tepe çıkarılır ise K_{n-1} elde edilir.
  Öyleyse $\kappa(K_n)=n-1$

Tanım 2: 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.

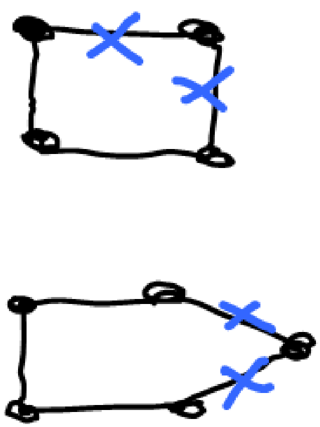
Öenmli grafların ayrıt connectivity değerleri:

1) Yol Graf



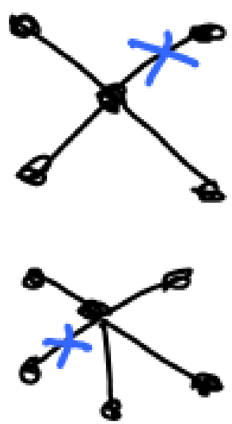
$k'(P_n) = 1$

2) Cevre Graf



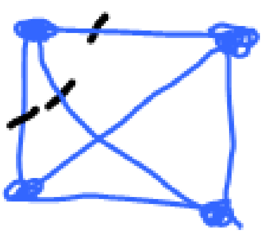
$k'(C_n) = 2$

3) Yıldız Graf



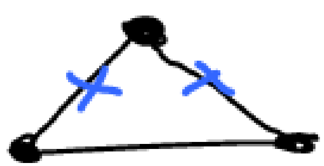
$\kappa'(K_{1,n}) = 1$

4) Tam Graf



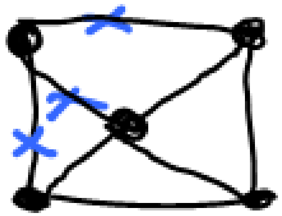
K_4

$\kappa'(K_N) = n-1$



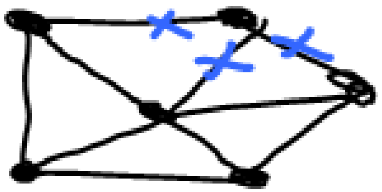
K_3

5) Tekerlek Graf



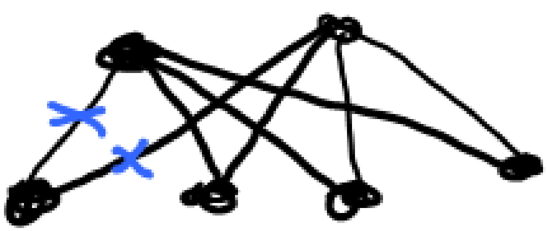
$W_{1,4}$

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



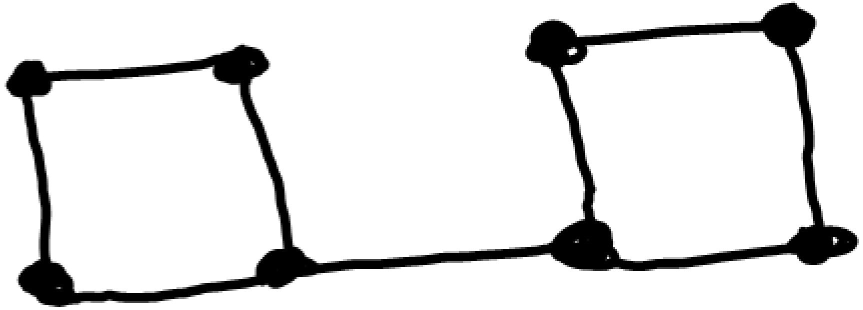
$W_{1,5}$

6) iki parçalı tam graf



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

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



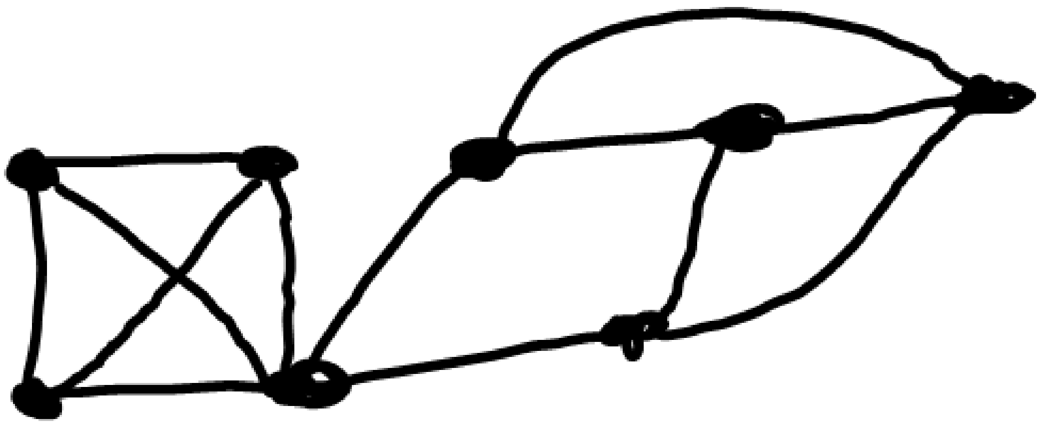
G

$\delta(G) = 2$

$k(G) = 1$

$k'(G) = 1$

$\delta(G) > k'(G) = k(G)$



G

$\delta(G) = 3$

$\kappa(G) = 1$

$\kappa'(G) = 2$

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

Kanıt. $\delta(G)$ minimum dereceli tepeye bitişik olan ayrıtlar, bir ayrıt kesim küme oluşturur. Buradan $\lambda(G) \leq \delta(G)$ sonucuna ulaşılır. n tepeli herhangi bir grafın tepe bağlantılılığı, tam grafın bağlantılılığı gözönüne alınarak $\kappa(K_n) = n - 1$ ile sınırlansın. $G = (V, E)$ en az iki tepeli bir graf ve minimal ayrıt kesim kümenin, S tepe kümesini, diğer tüm tepelerin oluşturduğu $\overline{S} = V \setminus S$ kümesinden ayırır. S ve $\overline{S}$ arasındaki tüm ayrıtlar G'de ise,

$$\lambda(G) = | S | \cdot | \overline{S} | \geq | V | -1$$

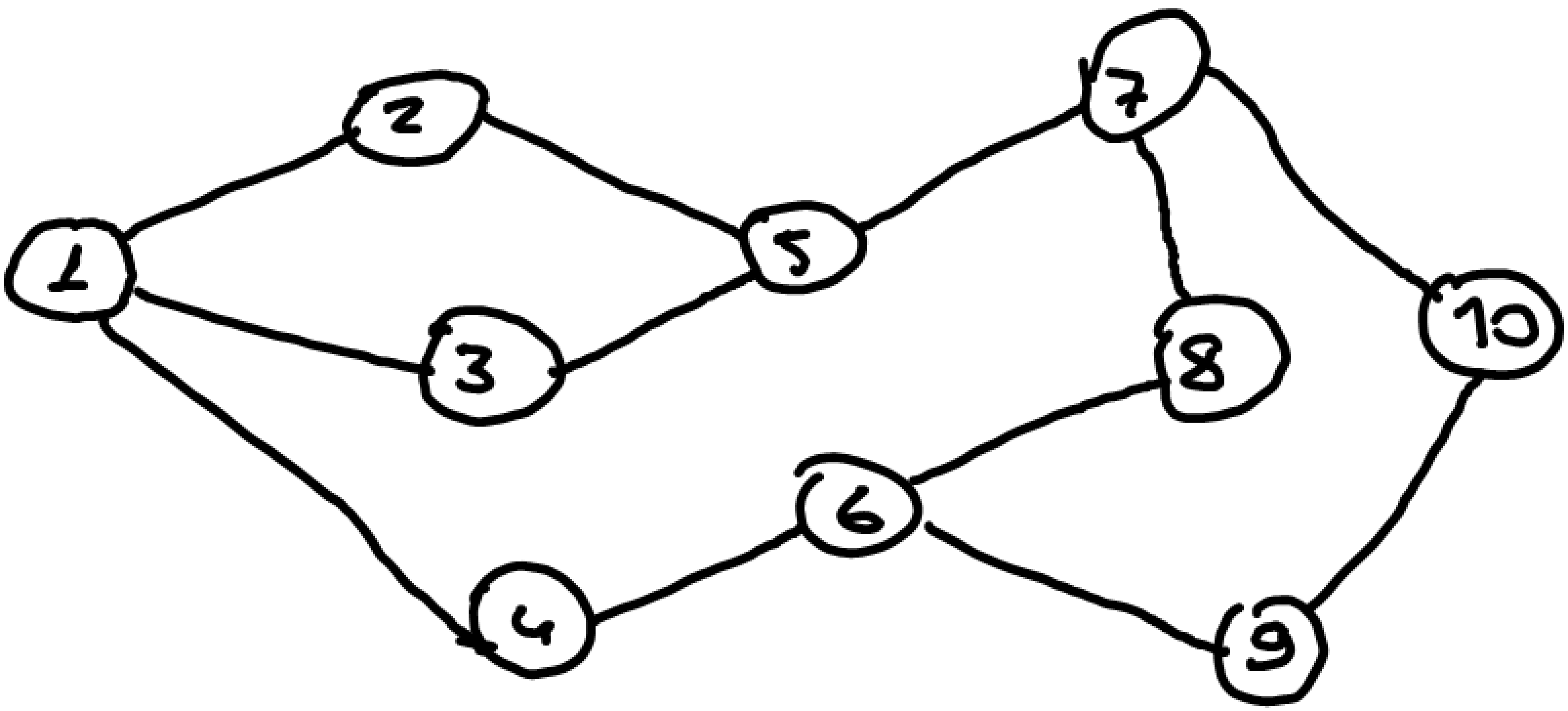
elde edilir. Aksi halde $\{x, y\} \notin E$ olan $x \in S$ ve $y \in \overline{S}$ tepeleri mevcuttur. x'in, $\overline{S}$'deki tüm komşularının kümesi ve buna ek olarak $S \setminus \{x\}$'deki tepelerden $\overline{S}$'da komşusu olanlar bir tepe kesim oluştururlar. Kesim kümenin eleman sayısı S'den $\overline{S}$'ye doğru olan ayrıtların maksimum sayısı kadardır ve bu kesim (en azından) x ve y'yi birbirinden ayırır. $\square$

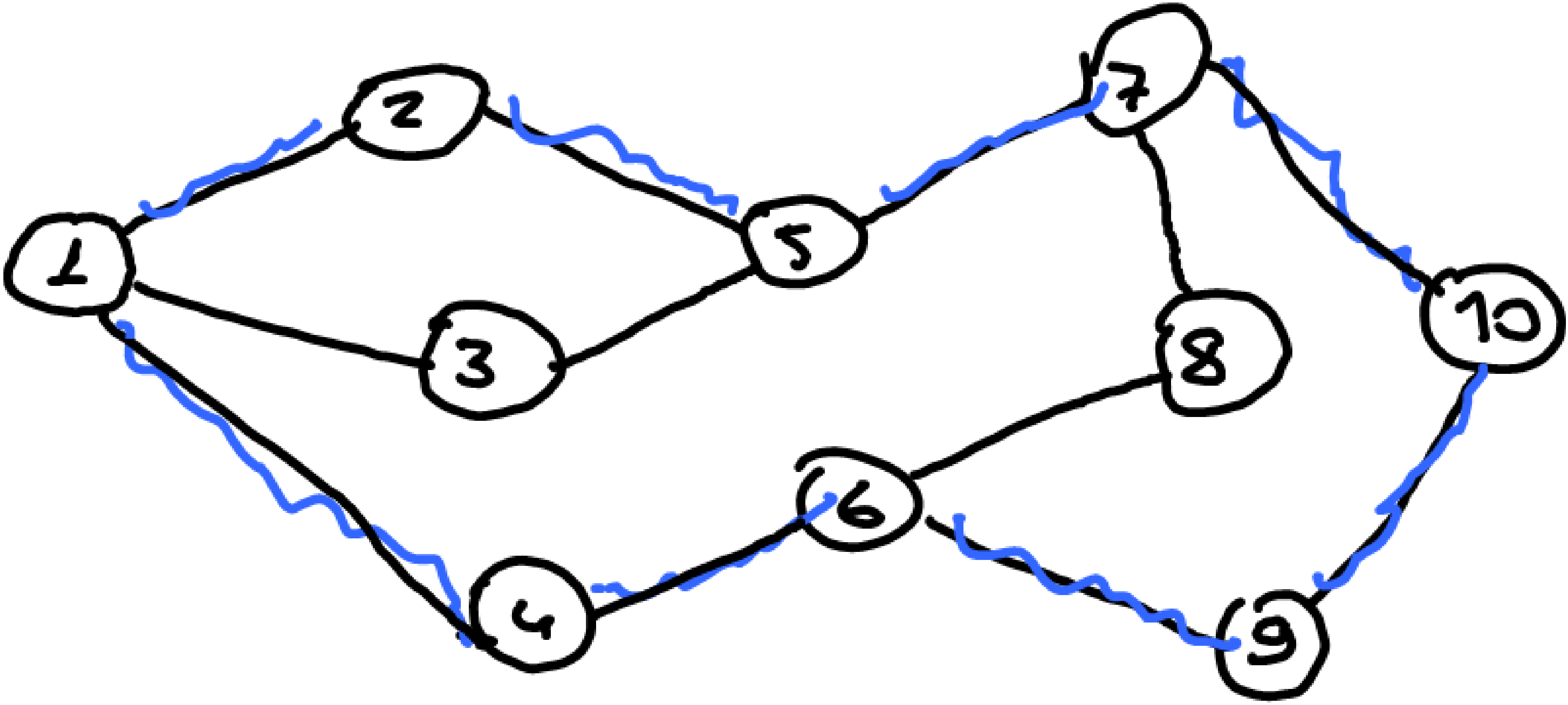
Bir Grafın Tepe ve Ayrıt Connectivity Değerlerinin Bulunması

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. Tepe birleştirilmişlik için ifade edilen bu teorem ayrıt tekrarsız yollar kullanılarak ayrıt birleştirilmişlik için de geçerlidir. 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 } yolları 5 ve 6

1,4,6,9,10 } tepelerini kullanan

içten ayrık yollardır.

5 ve 6 tepelerini kullanan farklı

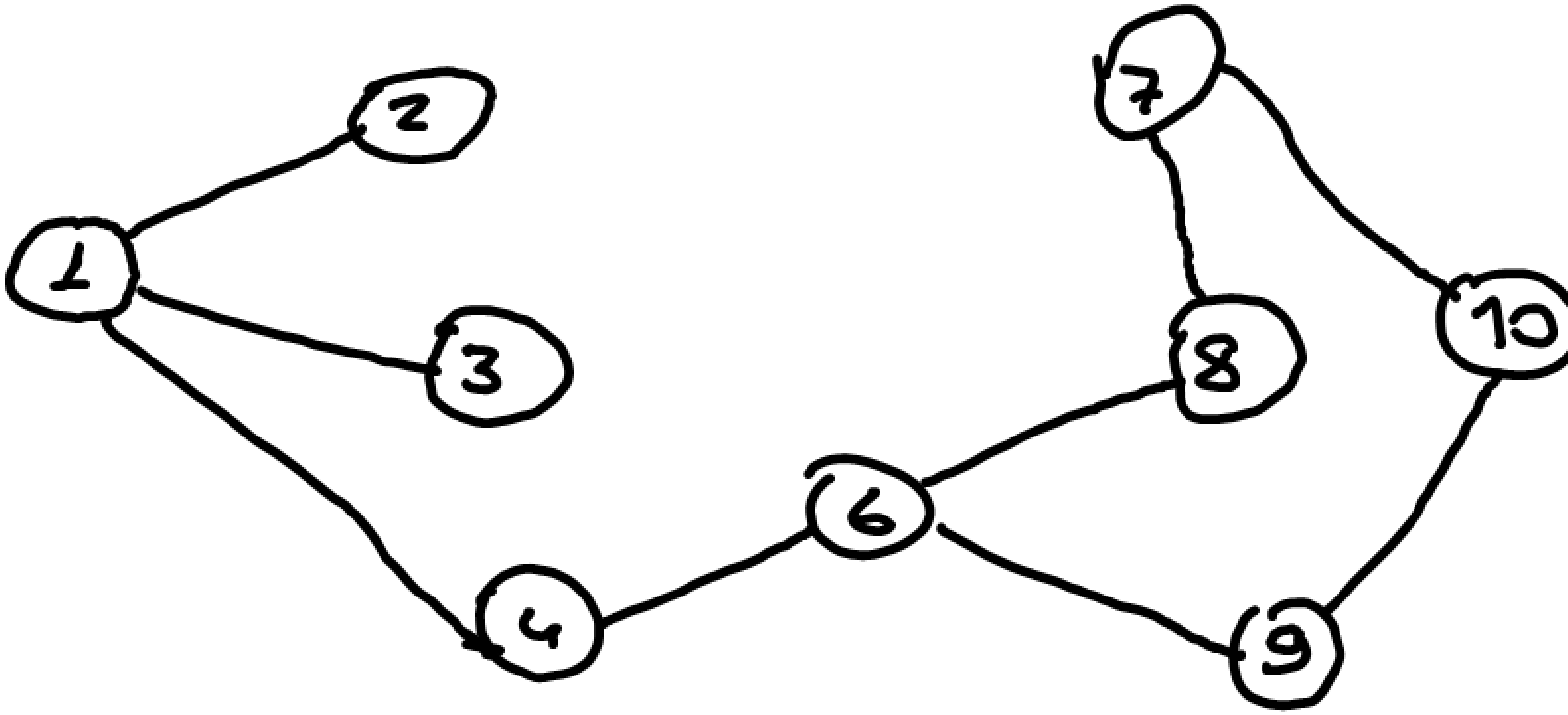
içten ayrık yollarda bulunabilir.

Fakat bunların sayısı maksimum

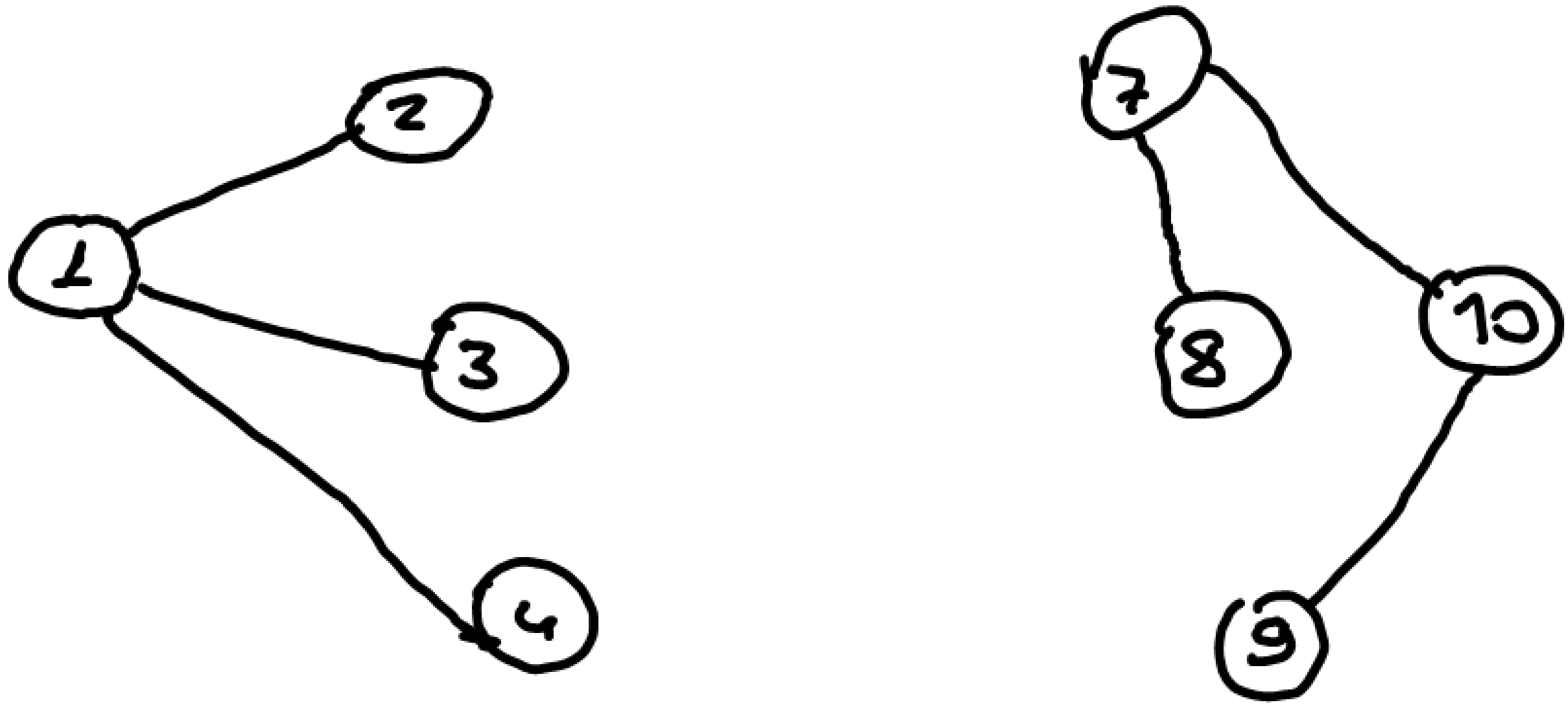
2'dir.

Böylece $k(G) = 2$ elde edilir.

5 tepesi silindiğinde



5 ve 6 tepeleri silindiğinde:



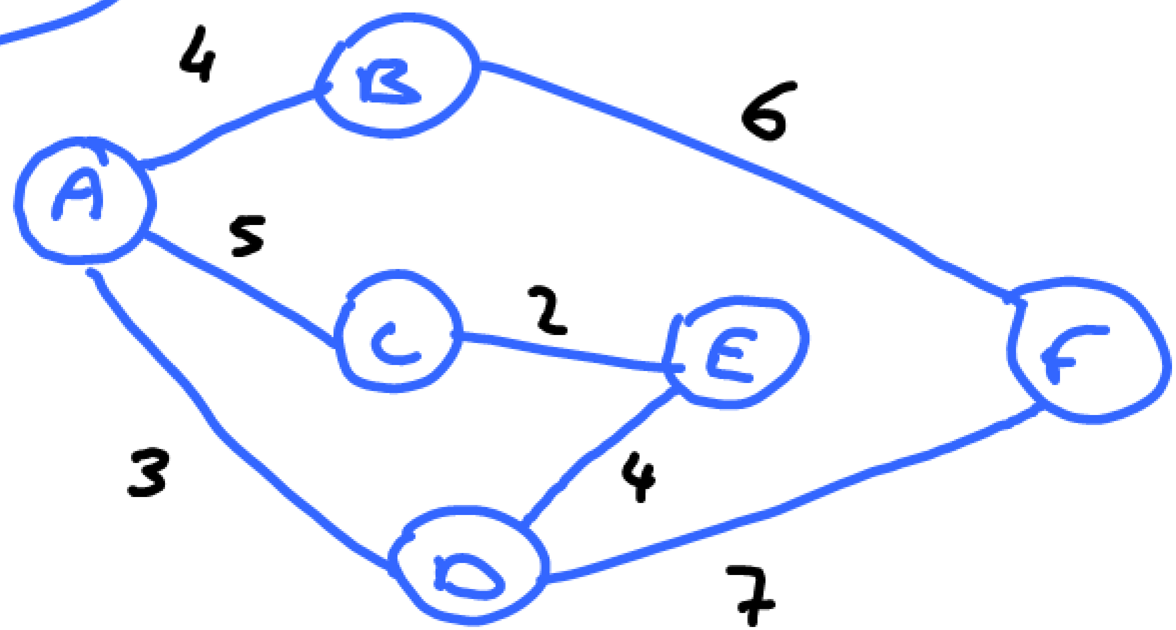
* içten ayrık yolların sayısı Ford-Fulkerson (en büyük akış) algoritması yardımıyla bulunabilir.

Ford-Fulkerson Sözde Kodu

```
ford-fulkerson (G, s, t)
for each edge (u, v) ∈ 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) | (u, v) ∈ 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)

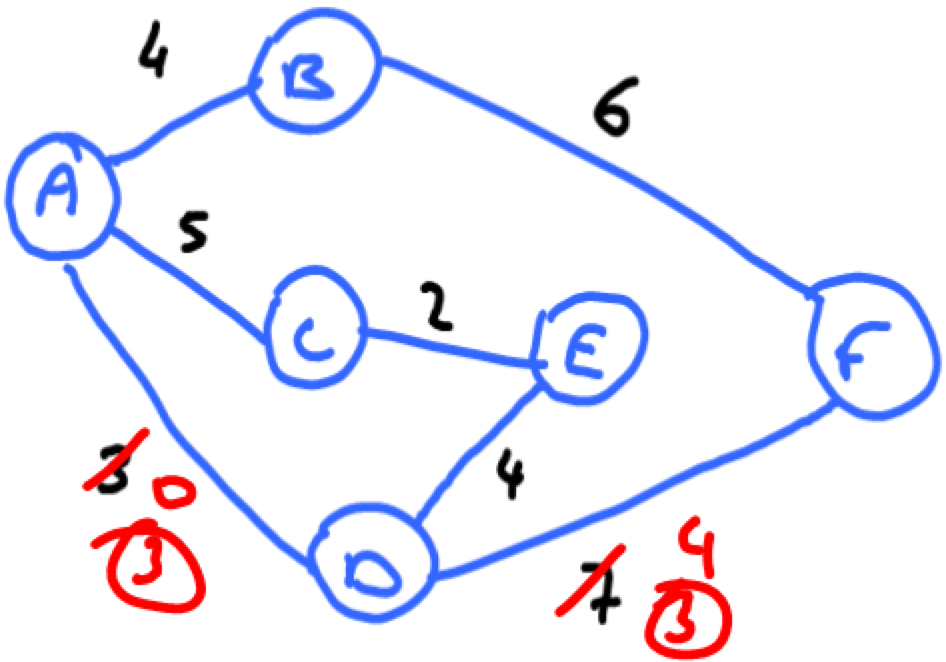


A→F'ye en büyük akış?

1. adım : DFS ile A→F ye yollar bulunur.

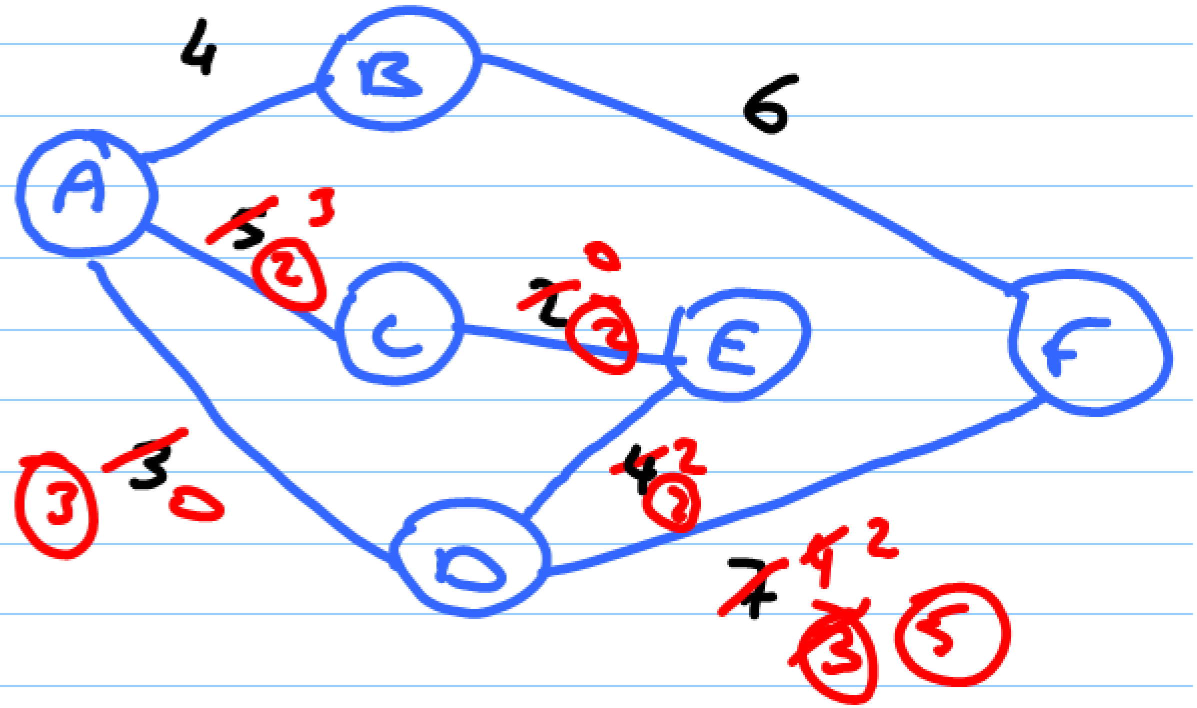
ADF yolu

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



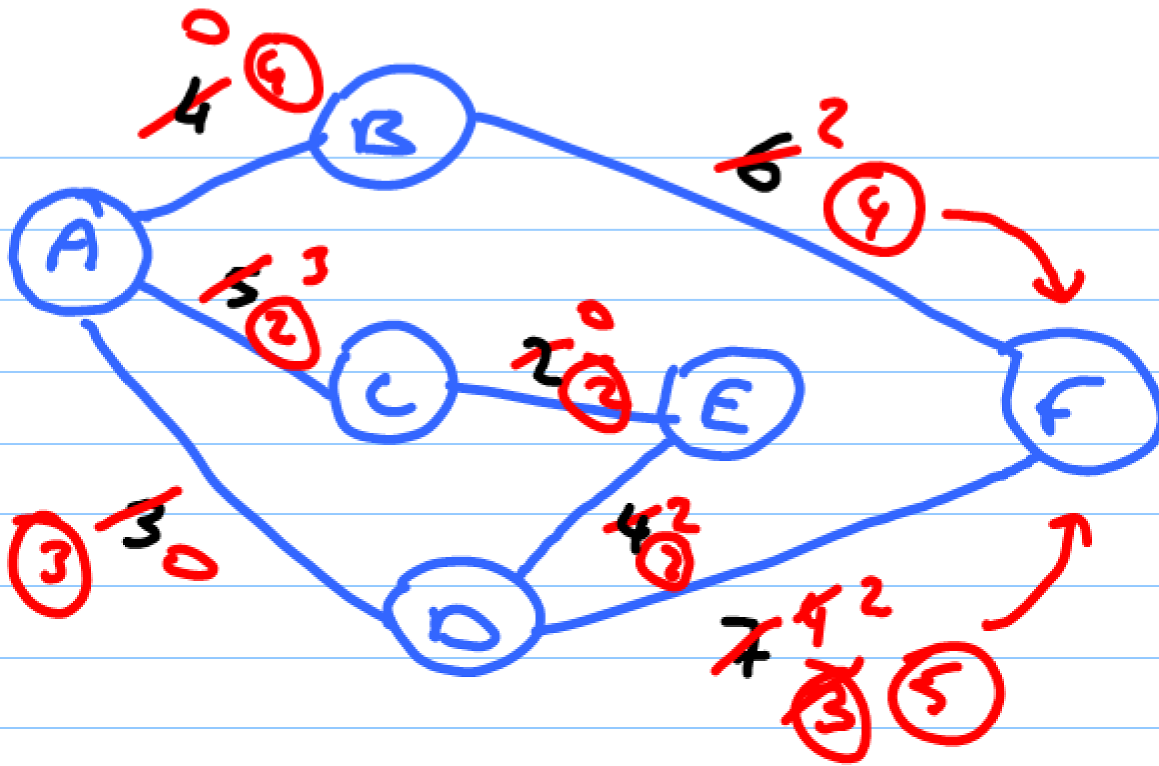
2. adım: A C E D F yolu

$$\min\{\underset{5}{A-C},\ \underset{2}{C-E},\ \underset{4}{E-D},\ \underset{3}{D-F}\} = 2$$



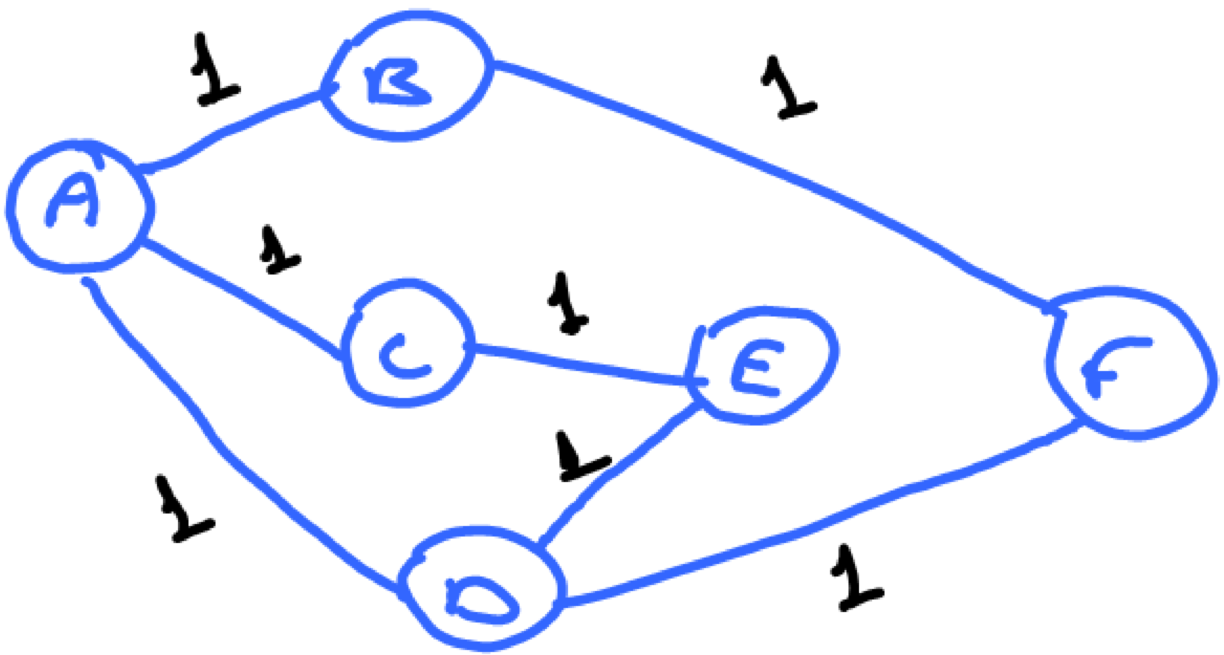
3. adım

ABF yolu $\quad \min\{\underset{4}{A\to B},\ \underset{6}{B\to F}\} = 4$



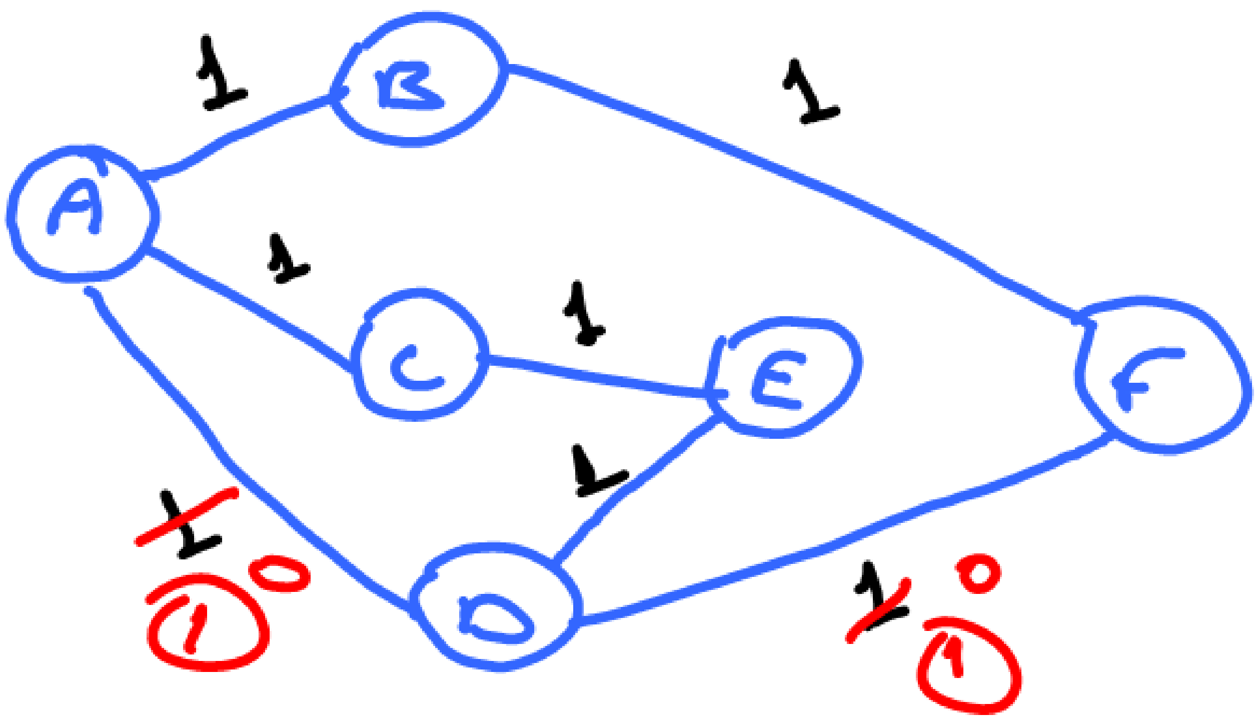
A'dan F'ye $4+5 = 9$ birim akış oldu.

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



1. adım: ADF yolu

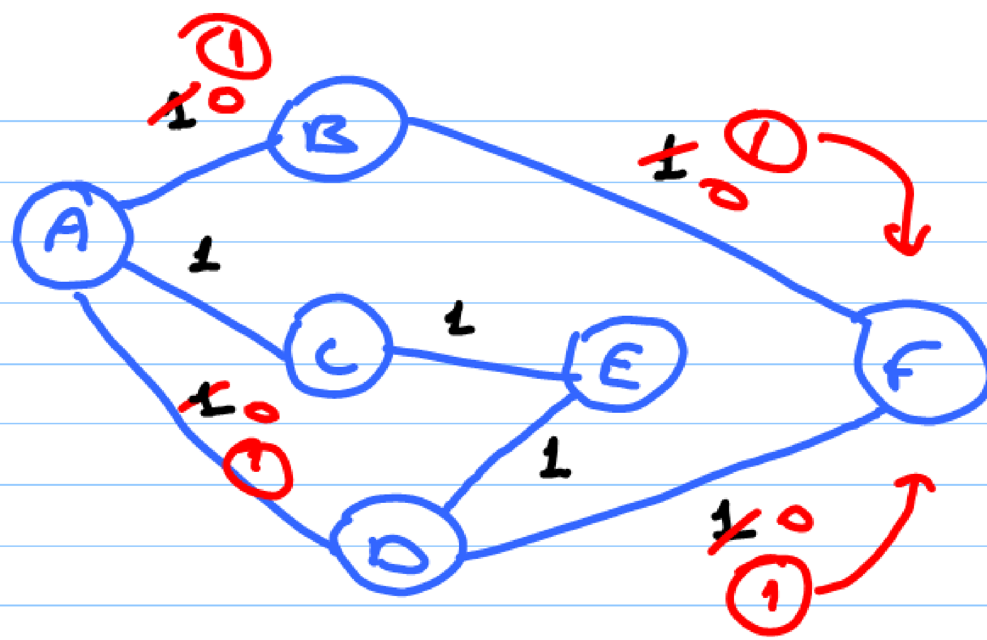
$$\min\{\underset{1}{A \to D}, \underset{1}{D \to F}\} = 1$$



2. adım: ACEDF yolu

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

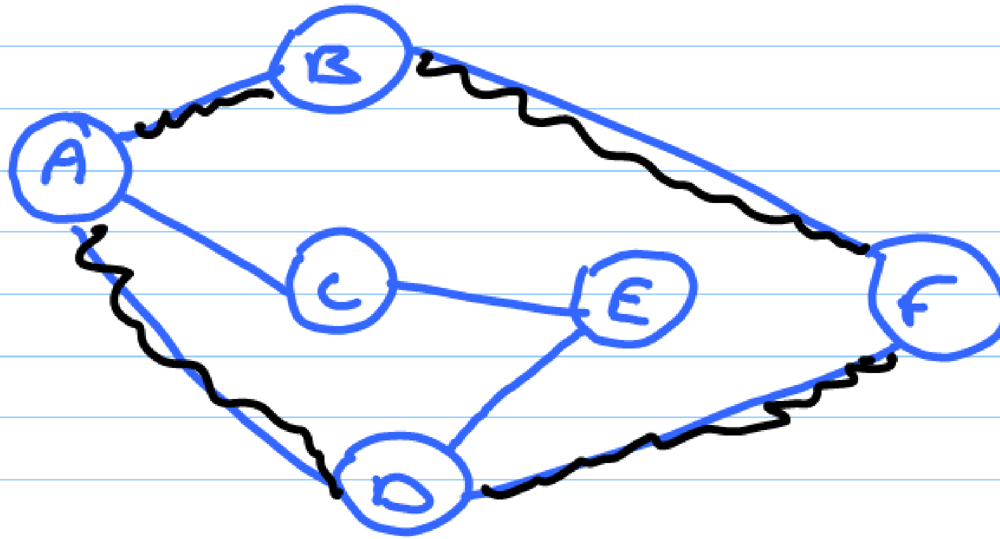
3. adım) ABF yolu $\min\{\underset{1}{A-B},\ \underset{1}{B-F}\} = 1$



$1+1=2$ 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!!!

Bir Grafın Tepe ve Ayrıt Connectivity Değerlerinin Bulan Alternatif Bir Algoritma

Tepe Geçiş Yoğunluğu

Grafın tüm tepe çiftleri arasındaki en kısa yolların üzerinde bulunan *tepelerin kullanılma sıklığına* denir.

Kritik Tepe

Grafta tepe geçiş yoğunluğu en büyük olan tepe veya tepelere denir.

Ayrıt Geçiş Yoğunluğu

Grafın tüm tepe çiftleri arasındaki en kısa yolların üzerinde bulunan *ayrıtların kullanılma sıklığına* denir.

Kritik Ayrıt

Grafta ayrıt geçiş yoğunluğu en büyük olan ayrıt veya ayrıtlara denir.

Algoritma:

i) Çizgenin bitişiklik matrisini oku

ii) $\kappa = 0$ / $\lambda = 0$ al

iii) Floyd algoritması ile çizgenin bağlantılı olup olmadığını test et,

tepe / ayrıt geçiş yoğunluğunu hesapla.

iv) Eğer çizge bağlantılı değilse adım vii)'e git

v) Kritik tepeyi / kritik ayrıtı çizgeden sil

$\kappa = \kappa + 1$ / $\lambda = \lambda + 1$ işlemini yap

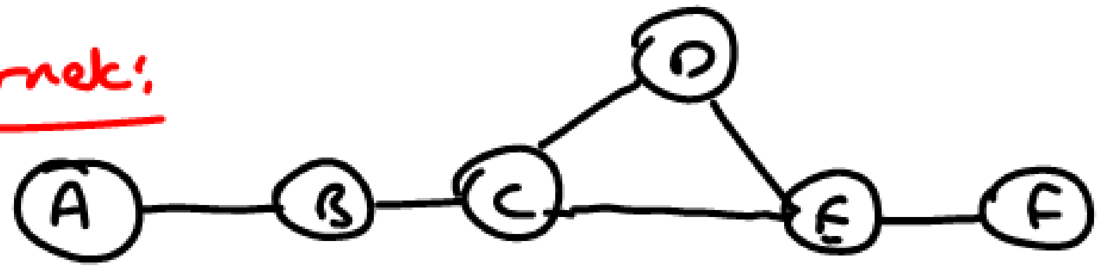
vi) Adım iii) ' e git

vii) Tepe Birleştirilmişlik Sayısını κ

Ayrıt Birleştirilmişlik Sayısını λ yazdır.

viii) SON.

Örnek:



Tepe yoğunluk matrisi

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
A & - & 4 & 3 & 0 & 1 & 0 \\
B & 0 & - & 3 & 0 & 1 & 0 \\
C & 0 & 1 & - & 0 & 1 & 0 \\
D & 0 & 1 & 2 & - & 1 & 0 \\
E & 0 & 1 & 2 & 0 & - & 0 \\
F & 0 & 1 & 2 & 0 & 3 & - \\
\hline
+ & & & & & & \\
\text{Sütun toplamı} & 0 & 8 & 12 & 0 & 7 & 0 \\
 & & & * & & &
\end{array}
$$

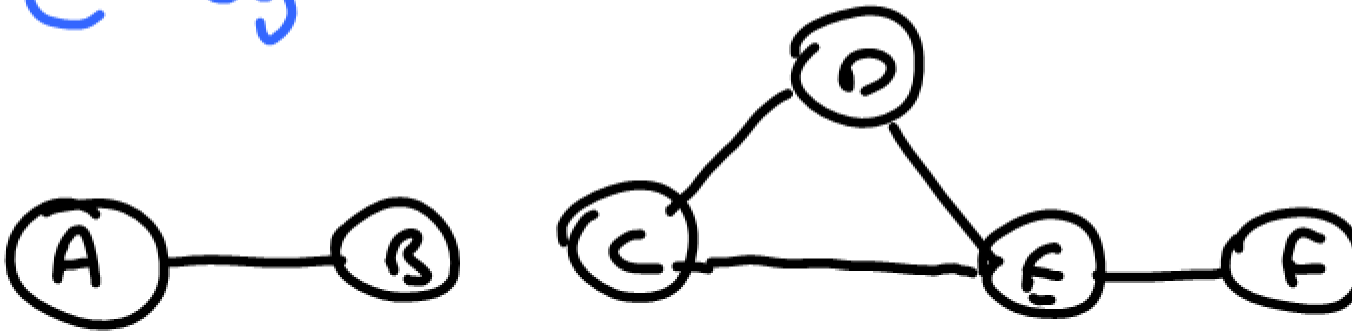
c, kritik tepedir.

c, tepesi çıkarıldığında Graf bağlantısız kalır. Böylece $k(G) = 1$ 'dir.

Ayrıt Yoğunluk Matrisi

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
A & - & 5 & 0 & 0 & 0 & 0 \\
B & 5 & - & \boxed{8} & 0 & 0 & 0 \\
C & 0 & \boxed{8} & - & 3 & 6 & 0 \\
D & 0 & 0 & 3 & - & 2 & 0 \\
E & 0 & 0 & 6 & 2 & - & 5 \\
F & 0 & 0 & 0 & 0 & 5 & -
\end{array}
$$

BC ayrıtı kritik ayrıttır.



BC ayrıtı çıkarıldığında Graf bağlantısız kalır. Böylece $k'(G) = 1$ 'dir.

FLOYD ALGORİTMASI

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.

```
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
```

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] M.E.Berberler, 2006. Çizgelerde Birleştirilmişlik Sayısı Üzerine.