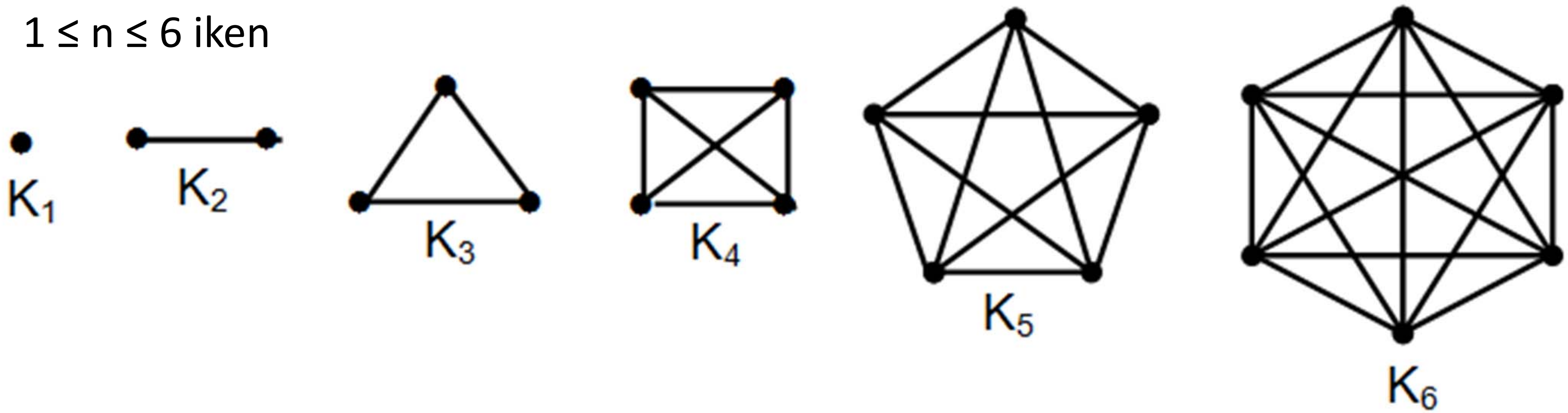
AYRIK MATEMATİK DERSİ

GRAF TEORİSİ
(ÇİZGELER-2)

Konya, 2020

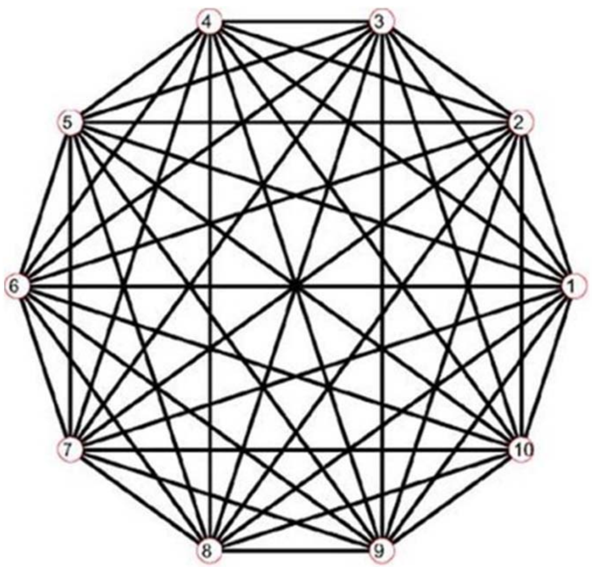
$1 \le n \le 6$ iken



Yönsüz bir grafta, bir "tam komple graf" (complete); farklı düğüm çiftlerinin tümü bir kavis ile bağlı olan basit graftır ve n düğüm sayısı olmak üzere K_n ile gösterilir.

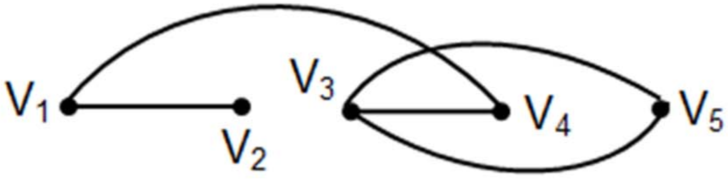
Düğüm sayısı n iken

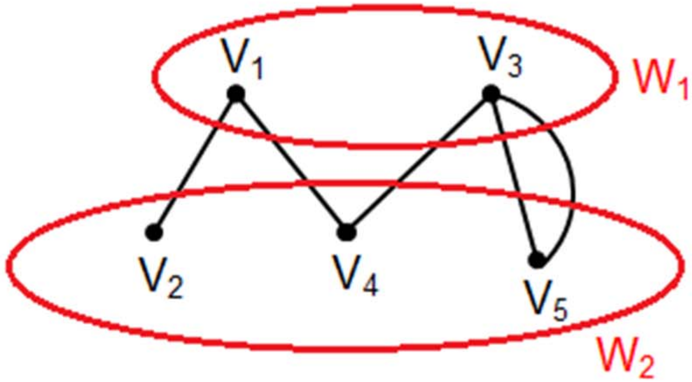
kavis sayısı = $\frac{n(n-1)}{2}$ olur. "Tam graf"; n düğüm sayısı iken r=(n-1). Dereceden düzenli graftır.



Platon graf; r dereceli düzenli grafların çok yüzlü olanlarına verilen genel isimdir.

Şekildeki n=10 ve r=9 olan graf



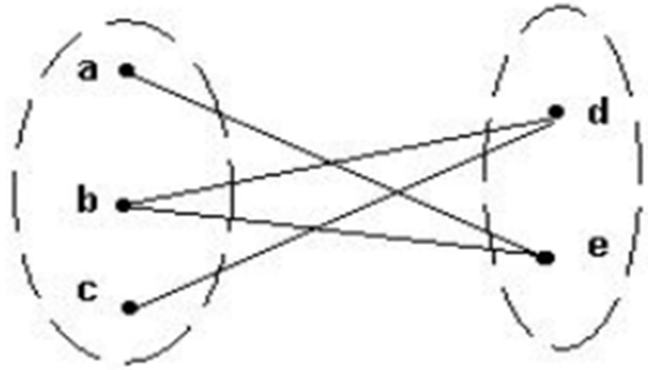


G yönsüz bir graf iken, V(G)={V_1, V_2, V_3……..V_n} ‘de V_1, V_2….V_n düğüm noktalarıdır. Tepe (düğüm) noktalarını W_1 ve W_2 gibi öylesine 2 kısma bölelim. Eğer:

1) W_1 kısımdan çıkan kavisler W_2’deki düğümlere gidiyorsa,
2) Hem W_1 kısımındaki düğümlerin kendi arasında hem de W_2 kısmındaki düğümlerin kendi arasında kavis yoksa
Bu graf “2 parçalı graftır.” (Bipartite Graph)

Iki Parçali (Bipartite) Graflar

- G, bipartite graf ise:
  - $V(G) = V(G_1) \cup V(G_2)$
  - $|V(G_1)| = m$, $|V(G_2)| = n$
  - $V(G_1) \cap V(G_2) = \emptyset$

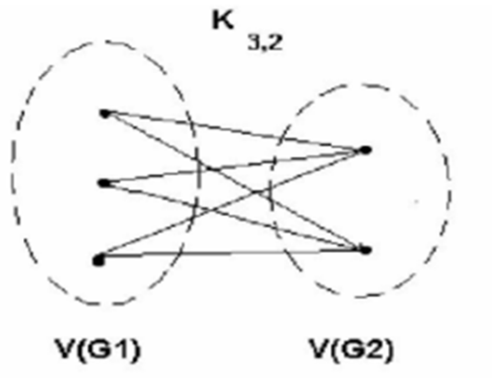


• Bir grafi olusturan dügümleri iki ayri kümeye bölerek grafi ikiye ayirabiliriz. Bu ayirma isleminde izlenecek yol; bir kenar ile birbirine baglanabilecek durumda olan dügümleri ayni küme içerisine yerlestirmemektir.

• Mevcut küme içerisindeki dügümler birbirlerine herhangi bir kenar ile baglanmamalidir.

2 parçalı grafta; eğer 1. Kısımdaki her düğüm, 2. Kısımdaki her düğümle ilişkide ise, “tam 2 parçalı graf” olur. Burada ; $K_{n,m}$ için n= 1. Kısımdaki düğüm sayısı ve m=2. Kısımdaki düğüm sayısıdır.

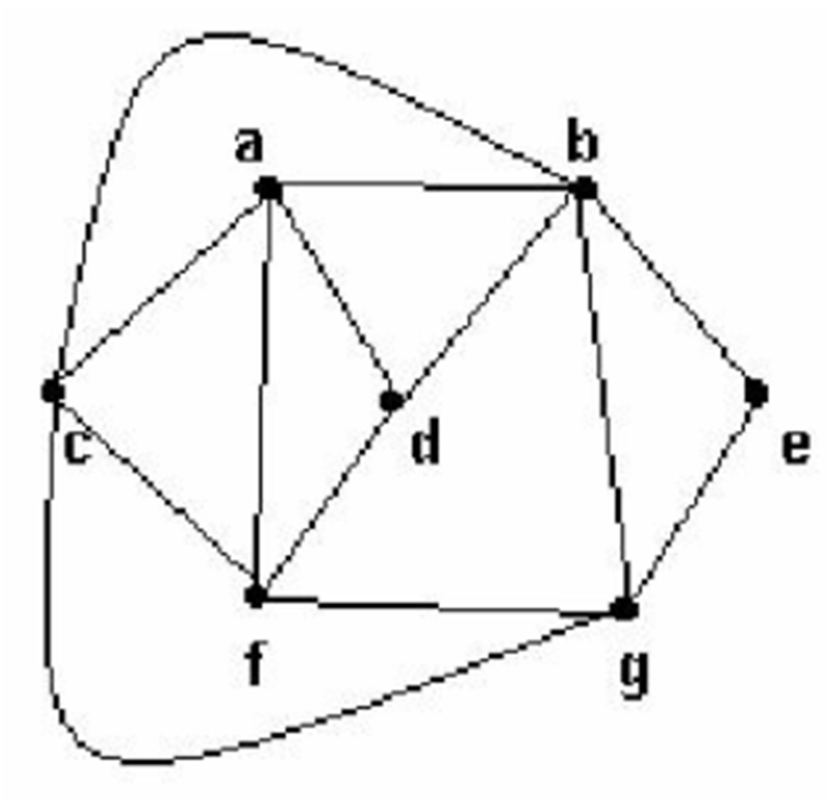
Tam (complete) bipartite graph $K_{m,n}$



- *complete* bipartite graf $K_{m,n}$ seklinde gösterilir. Ilgili grafin dügümlerinin kümesi m ve n elemanli iki alt kümeye ayrilir.
- Bir kenari birbirine baglayan iki dügümünde farkli alt kümelerin elemani olmak zorundadirlar.
- $|V(G_1)| = m$
- $|V(G_2)| = n$

PLANAR (YASSI) GRAF

Planar Graflar

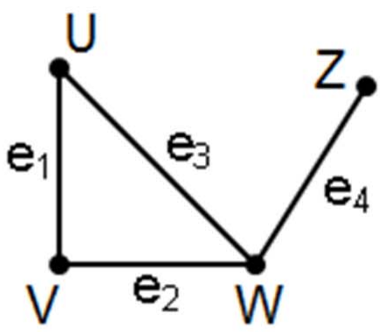


Bir G grafinin kenarlari birbirlerini kesmeyecek sekilde çizilebiliyorsa ***Planar*** graf olarak adlandirilir.

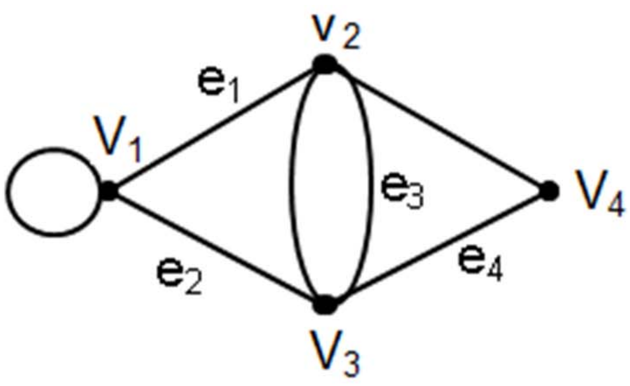
Alt graflar:

Yönsüz G ve H grafları için; H grafının tüm düğümleri V(G)'ye dahilse yani G grafının düğümleri kümesine dahilse ve H grafının tüm kavisleri E(G)'ye dahilse yani G grafının kavis kümesine dahilse;

$V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ ve için $\delta_H\,(E\,) = \delta_G\,(E\,)$ ise yani ayni kavisler aynı düğümleri birleştiriyorsa; H grafı G grafının alt grafıdır.



H



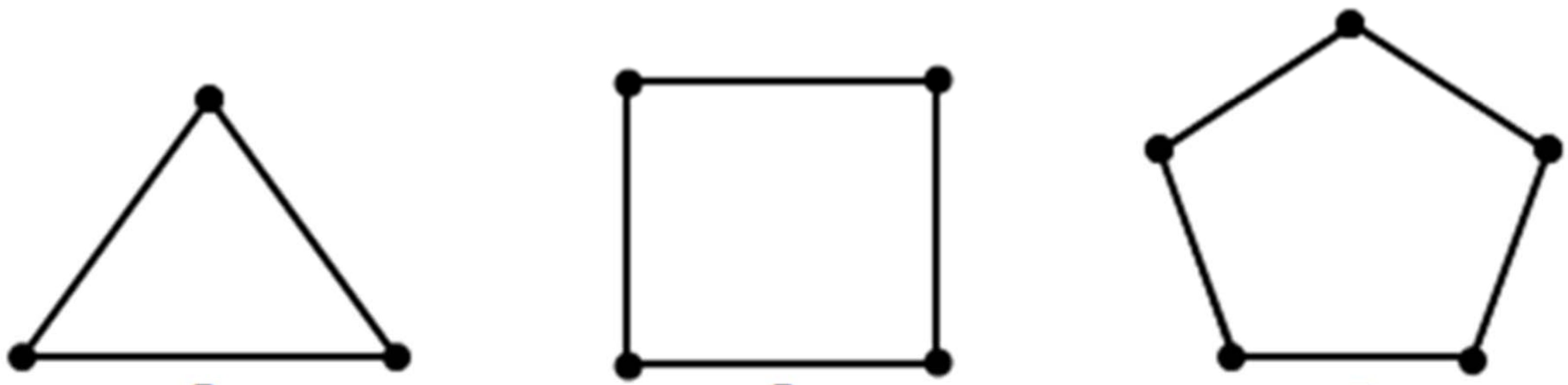
G

$U = V_2$
$V = V_1$
$W = V_3$
$Z = V_4$

H grafı G grafının bir alt grafıdır.

Cycles (Çember) Graf C_n

- $n \geq 3$
- *cycles graph* C_n : n adet dügüm ve $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$, dügüm çiftlerinden olusan kenarlardan meydana gelir.



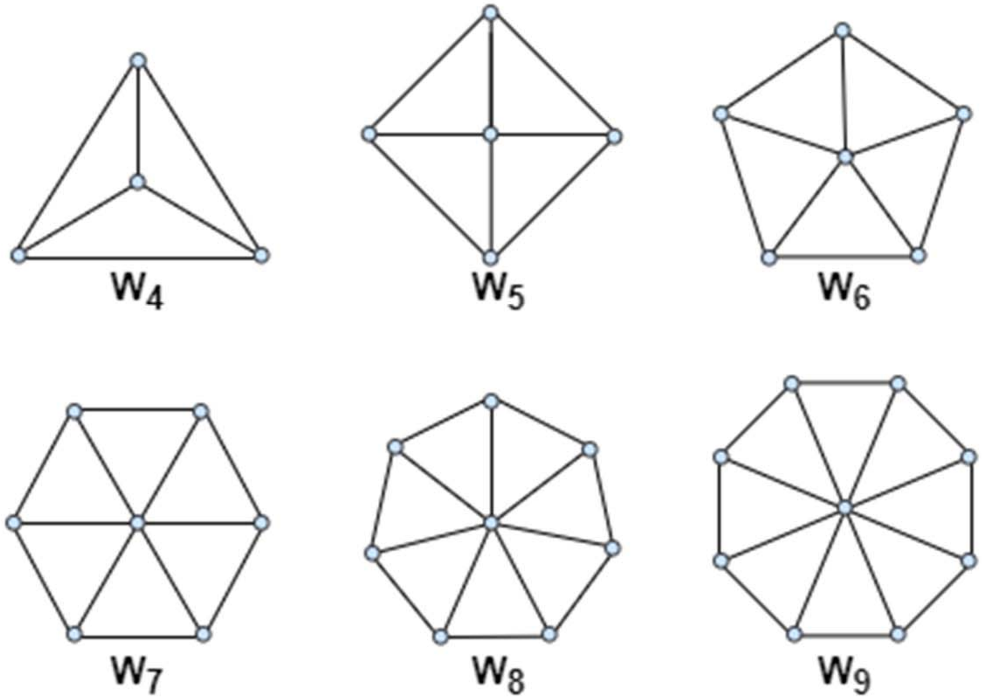
C_3 C_4 C_5

Wheel (Tekerlek) Graf W_n

- *wheel graph* W_n :

  Cycle C_n grafina ek bir dügüm eklenerek olusturulur.

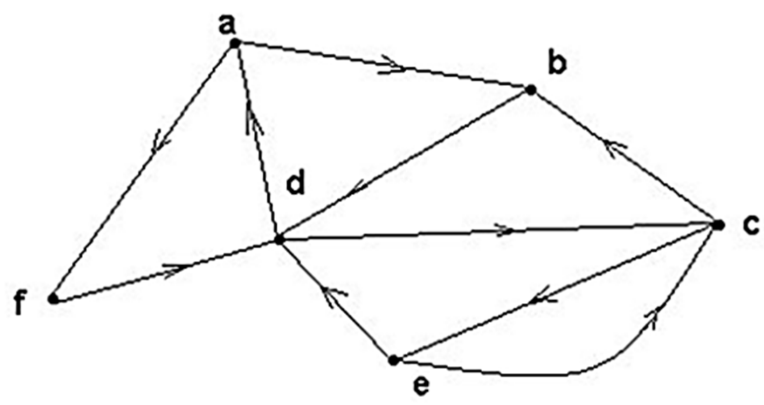
  Eklenen yeni dügüm, diger bütün dügümlere baglidir.



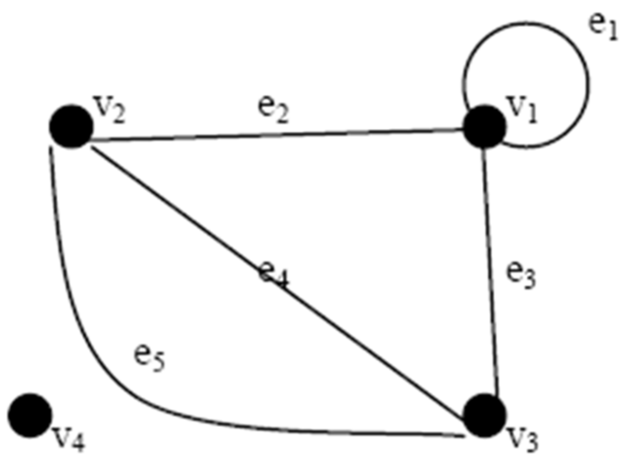
Yönlendirilmiş (Yönlü) ve Yönlendirilmemiş (Yönsüz) Graflar:

Bir grafta bulunan kenarların başlangıç-bitiş olarak yön bildirmemesi durumunda bu grafa “yönsüz graf” denilir. Bu durumda iki düğüm (tepe) arasında bulunan kenar (kavis), her iki yönlü de hareket edilebileceğini ifade eder.

Bazı durumlarda kavislerin bir tepeden çıkıp, diğerine gitmesinin gösterilmesi gerekir. Bu durum bir okla gösterilir. Eğer grafta kavislerin bu şekilde yönü varsa ve oklarla gösterilmişse buna “yönlendirilmiş ya da yönlü graf” denir.



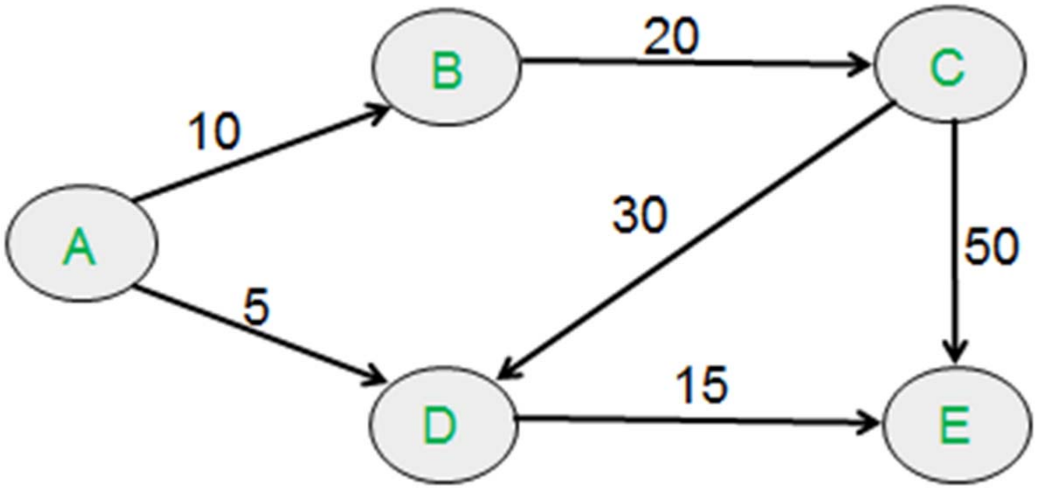
Yönlü Graf



Yönsüz Graf

<u>Ağırlıksız ve Ağırlıklı (maliyetli) graflar:</u>
Her bir kenarına nümerik(sayısal) bir değer yani ağırlık verilmiş bir graftır. Graf kenarları üzerinde ağırlıkları olabilir. Eğer kenarlar üzerinde ağırlıklar varsa bu tür graflara “ağırlıklı/maliyetli graf” denir.

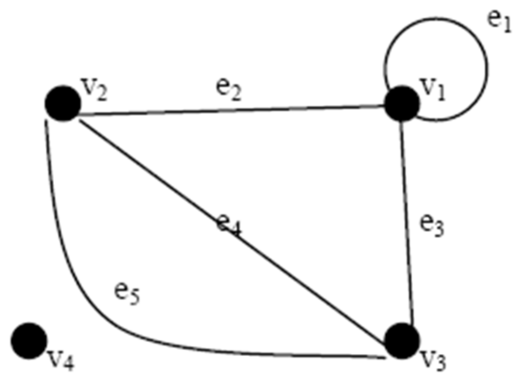
Eğer tüm kenarların maliyeti 1 veya birbirine eşitse bu tür graflara da “ağırlıksız/maliyetsiz graf” denir.



Yönlü ağırlıklı Graf

Yönsüz graflarda komşuluk matrisi (Adjacency matrice):

Yönsüz G grafı için, $V(G)=\{V_1, V_2, V_3.....V_n\}$ düğüm kümesi olsun. G'nin komşuluk matrisi V_i ve V_j düğümlerini bağlayan a_{ij} kavis sayısını gösteren A(G) matrisidir. A(G) matrisi, bir grafı matematiksel olarak bazı işlemlere tabi tutmak için kullanılır.



	V_1	V_2	V_3	V_4
V_1	1	1	1	0
V_2	1	0	2	0
V_3	1	2	0	0
V_4	0	0	0	0

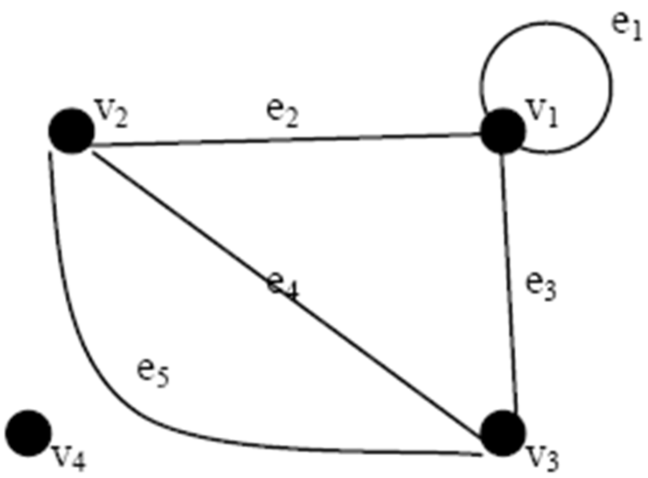
Bu matrise bakıldığında tek loop görünür o da $V_1 - V_1$ içindir. Düğümlerin derecelerine bakılırsa;

$\delta\ (V_1) = 2.1 + 1 + 1 = 4$ (2 looptan gelir)

$\delta\ (V_2) = 1 + 2 = 3$

$\delta\ (V_3) = 1 + 2 = 3$

$\delta\ (V_4) = 0$



	V_1	V_2	V_3	V_4
V_1	1	1	1	0
V_2	1	0	2	0
V_3	1	2	0	0
V_4	0	0	0	0

Yönsüz bir G grafında:

V_i'yi V_j'ye bağlayan kavis sayısı = V_j'yi V_i'ye bağlayan kavis sayısı

Bu nedenle komşuluk matrisi simetrik olur.

Yönsüz Graflarda komşuluk matrisi:

Yönsüz bir graf için; Graf komşuluk matrisi kullanılarak, matristeki kavis (kenar) sayısı bulunabilir. Bunun için:

1) Komşuluk matrisinde sol üst köşeye göre köşegen çizilir.

2) Komşuluk matrisi simetrik olduğundan, üst yada alt köşegen matrisi kullanılabilir.

3) Alt ya da üst köşegen matrisindeki sayılar (toplam1)toplanır.

4) İlk adımda çizilen ve loop'ları gösteren köşegen üstündeki sayılar toplanır (toplam2).

5) Köşegen matristeki toplam sayı (toplam1) ile köşegen üzerindeki sayıların toplamı (toplam2), graf üzerindeki toplam kavis sayısını verir.

	V_1	V_2	V_3	V_4
V_1	1	1	1	0
V_2	1	0	2	0
V_3	1	2	0	0
V_4	0	0	0	0

Kırmızı alandaki sayıların toplamı = 1+1+2 = 4

Köşegen üzerindeki sayıların toplamı = 1

Toplam 4+1=5

YA DA

Mavi alandaki sayıların toplamı = 1+1+2 = 4

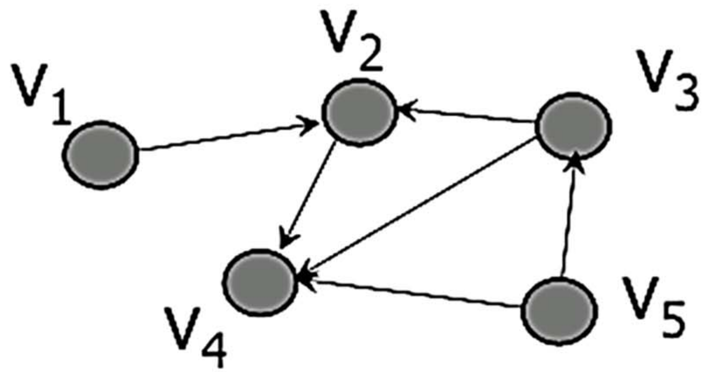
Köşegen üzerindeki sayıların toplamı = 1

Toplam 4+1=5

Yönlü Graflarda komşuluk matrisi:

Yönlü graflar için de komşuluk matrisi, yönsüz graflardaki mantık ile oluşturulur. Ancak yönlü G grafı için, $V(G)=\{V_1, V_2, V_3.....V_n\}$ düğüm kümesi iken, G'nin komşuluk matrisine başlangıç V_i ve bitiş V_j düğümleri olmak üzere (V_i, V_j) düğümleri arasında kavis varsa 1, yoksa 0 yazılır.

$$\text{Matris}[i][j] = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E \end{cases}$$



G

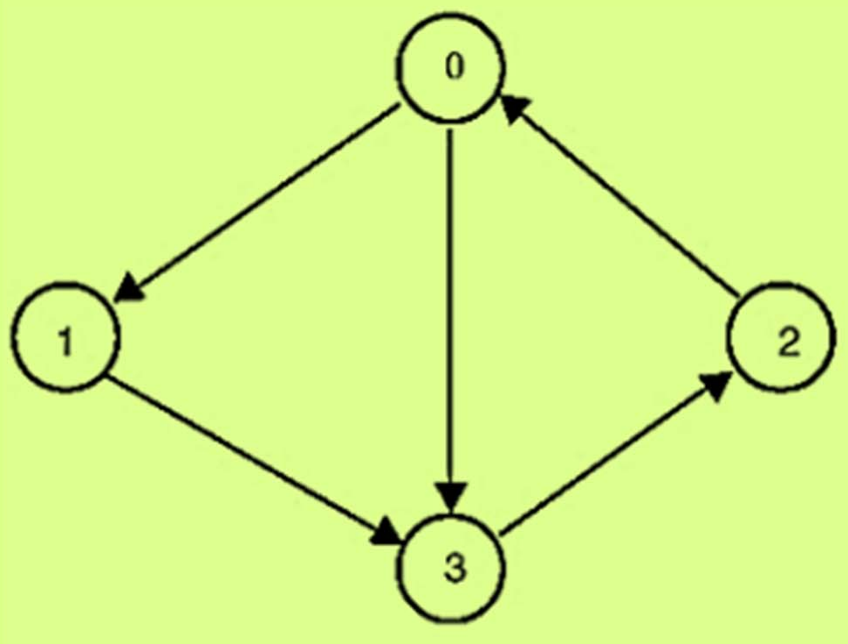
	V_1	V_2	V_3	V_4	V_5
V_1	0	1	0	0	0
V_2	0	0	0	1	0
V_3	0	1	0	1	0
V_4	0	0	0	0	0
V_5	0	0	1	1	0

Yönlü bir graf için, Graf komşuluk matrisi kullanılarak, matristeki kavis (kenar) sayısı bulunabilir. Bunun için, matristeki tüm elemanların sayısal değerleri toplanır.

Yönlü Graflarda düğüm derecesi:

Yönlü graflarda, düğüm derecesi giriş derecesi (input degree) ve çıkış derecesi (output degree) olarak ayrı ayrı belirtilir. Yönlü grafta bir düğüm için satırdaki 1'ler out-degree, sütundakiler ise indegree değerini verir. Bu ikisinin toplamı düğümün derecesidir.

- Indegree (girenler) ve outdegree (çıkanlar) matris üzerinde gösterimi



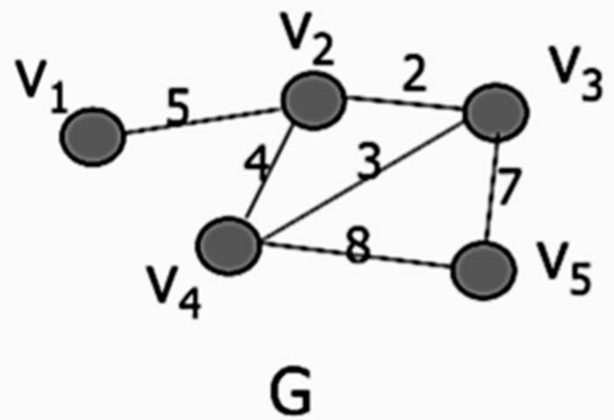
Graph G_1

	0	1	2	3
0	0	1	0	1
1	0	0	0	1
2	1	0	0	0
3	0	0	1	0

Ağırlıklandırılmış Graflarda komşuluk matrisi *:

Ağırlıklandırılmış yönsüz graflar için de komşuluk matrisi, yönsüz graflardaki mantık ile oluşturulur. Yönsüz G grafı için, $V(G)=\{V_1, V_2, V_3.....V_n\}$ düğüm kümesi olsun. G'nin komşuluk matrisi V_i ve V_j düğümlerini bağlayan a_{ij} kavisinin ağırlık değerini gösteren B(G) matrisidir.

$$\text{Matris}[i][j] = \begin{cases} w(v_i, v_j) & \text{if } (v_i, v_j)\in E \text{ or } (v_j, v_i)\in E \\ \infty & \text{otherwise} \end{cases}$$

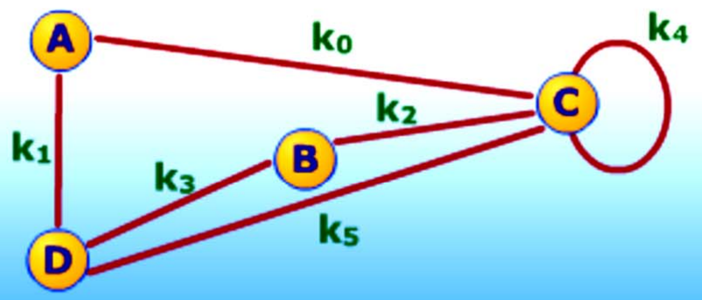


	V_1	V_2	V_3	V_4	V_5
V_1	∞	5	∞	∞	∞
V_2	5	∞	2	4	∞
V_3	∞	2	∞	3	7
V_4	∞	4	3	∞	8
V_5	∞	∞	7	8	∞

*: Konu ile ilgili kod yazımı ve örnek çalışmalar algoritmalar dersinde anlatılacaktır.

<u>Graf bitişiklik matrisi (Incedence matrice):</u>

Yönsüz bir G grafında, düğümlerle kenarlar arasındaki bağlantıyı yani bitişiklik ilişkisini gösteren bir matristir. Matrisin satır sayısı graftaki düğüm sayısı, sütun sayısı ise graftaki kenar sayısı kadar olur. Eğer satırdaki bir düğüme, sütundaki ilgili kavis bağlanmışsa 1, aksi durumda ise 0 yazılır.



a) Bir Graf

	A	B	C	D
A	0	0	1	1
B	0	0	1	1
C	1	1	1	1
D	1	1	1	0

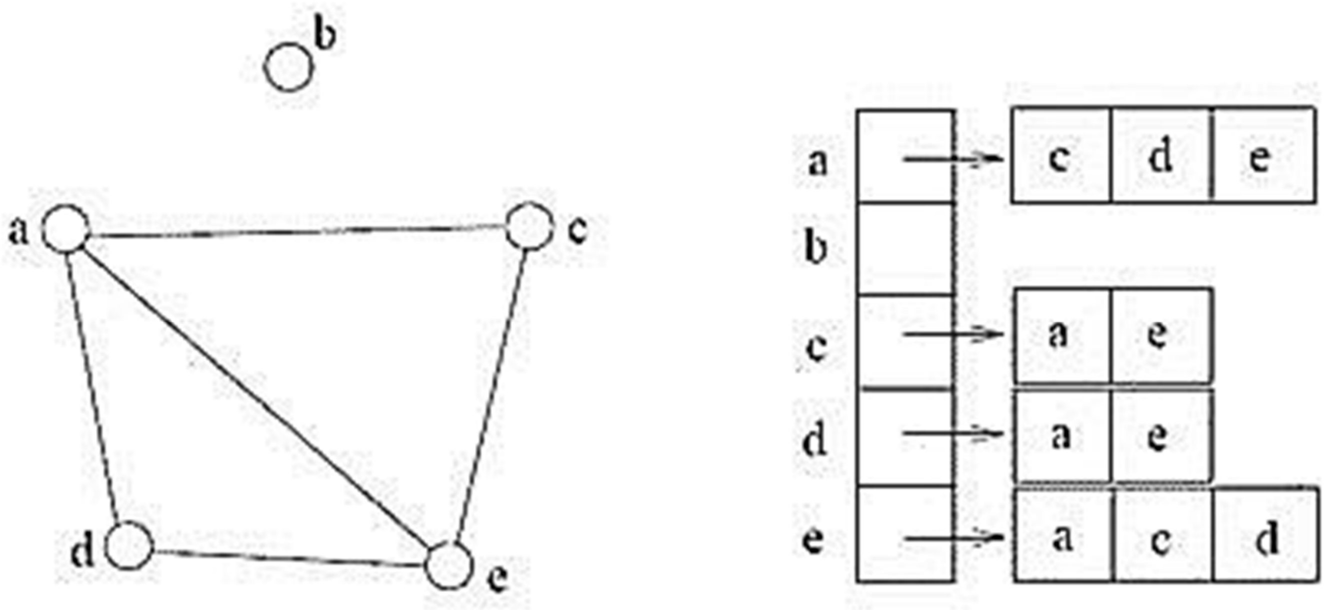
b) Komşuluk Matrisi

	k_0	k_1	k_2	k_3	k_4	k_5
A	1	1	0	0	0	0
B	0	0	1	1	0	0
C	1	0	1	0	1	1
D	0	1	0	1	0	1

c) Bitişiklik Matrisi

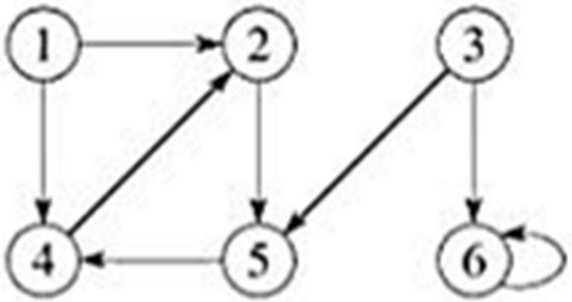
Graf komşuluk listesi *:

Yönsüz bir G grafı için komşuluk listesi:

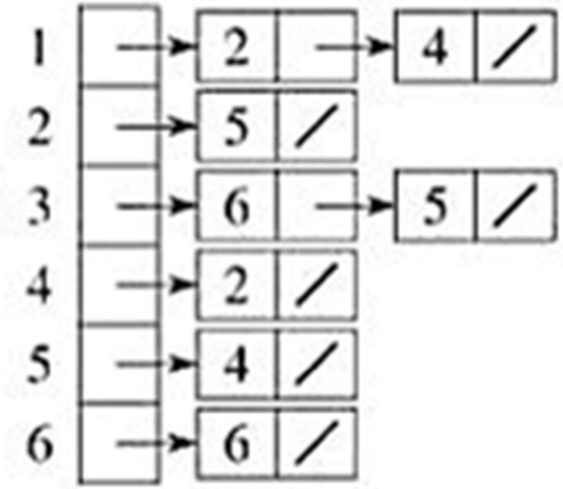


Graf komşuluk listesi*:

Yönlü bir G grafı için komşuluk listesi:



Komşuluk Listesi

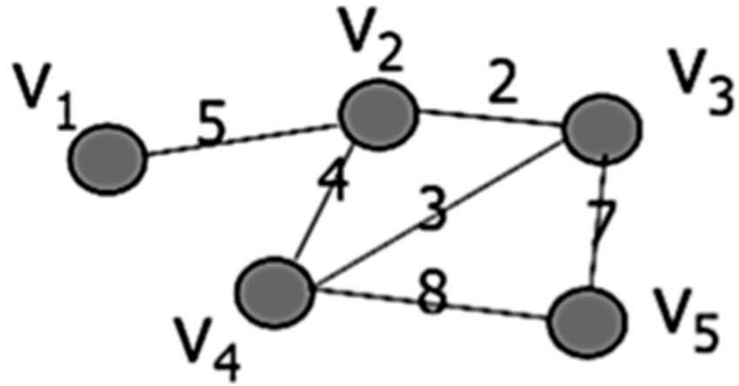


Komşuluk Matrisi

	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Graf komşuluk listesi*:

Yönsüz ve ağırlıklı bir G grafı için komşuluk listesi:



G

$v_1 \rightarrow v_2(5)$

$v_2 \rightarrow v_1(5) \rightarrow v_3(2) \rightarrow v_4(4)$

$v_3 \rightarrow v_2(2) \rightarrow v_4(3) \rightarrow v_5(7)$

$v_4 \rightarrow v_2(4) \rightarrow v_3(3) \rightarrow v_5(8)$

$v_5 \rightarrow v_3(7) \rightarrow v_4(8)$

Komşuluk matrisi ile komşuluk listesi karşılaştırması:

Avantaj ve dezavantajları:

Komşuluk matrisi:

1) Çok fazla hafıza alanına ihtiyaç duyar. Daha az hafızaya gerek olması için sparse (seyrek matris, içinde 0 olan matris) matris tekniklerinin kullanılması gerekir.

2) Herhangi 2 düğümün komşu olup olmadığı çok kısa sürede ve kolaylıkla tespit edilebilir.

Komşuluk listesi:

1) Bir düğümün tüm komşularına hızlı bir şekilde ulaşılır.

2) Daha az hafıza alanına ihtiyaç duyar.

3) Listenin oluşturulması, matrise göre daha zor olabilir.