

## Graflar (Graphs)

---

- Graf gösterimi
- Uygulama alanları
- Graf terminolojisi
- Depth first dolaşma
- Breadth first dolaşma
- Topolojik sıralama

---

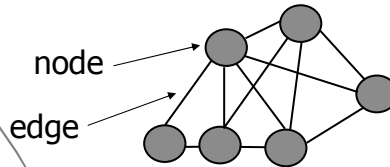
Yrd.Doç.Dr. M. Ali Akcayol

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar

---

- Graflar bilgi parçaları arasındaki ilişkileri gösterirler.
- Bir  $G$  graf  $V$  ile gösterilen node'lardan (verteks) ve  $E$  ile gösterilen kenarlardan (Edge) oluşur. Her kenar iki node'u birleştirir.
- Her node bir bilgi parçasını gösterir.
- Her kenar iki bilgi arasındaki ilişkiyi gösterir ve  $(u, v)$  şeklinde ifade edilir.  $(u, v)$  iki node'u gösterir.



---

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar

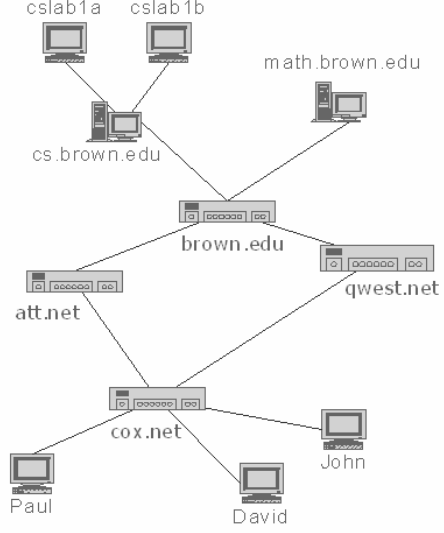
### Uygulama alanları

**Elektronik devreler**  
Baskı devre kartları (PCB)  
Entegre devreler

**Ulaşım ağları**  
Otoyol ağı  
Havayolu ağı

**Bilgisayar ağları**  
Lokal alan ağları  
internet

**Veritabanları**  
Entity-relationship diyagramı

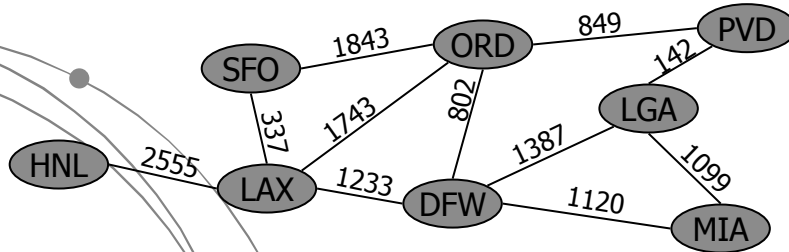


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar

### Örnek gösterim:

- Aşağıdaki şekilde her bir node üç karakter kodlanmış olarak bir havaalanını göstermektedir.
- Her bir kenar iki node arasındaki uçuş rotasını ifade etmekte ve iki node arasındaki uzaklığı ifade etmektedir.



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar

### Kenar türleri:

- **Directed edge (Yönlendirilmiş kenar)**
  - Sıralı node çiftleriyle ifade edilir.  $(u, v)$  ile  $(v, u)$  aynı değildir.
  - İlk node orijin ve ikinci node ise hedef olarak adlandırılır.
  - Örnek: iki nokta arasındaki uçuş.
- **Undirected edge (Yönlendirilmemiş kenar)**
  - Sırasız node çiftleriyle ifade edilir.  $(u, v)$  ile  $(v, u)$  aynı şeyi ifade ederler.
  - Örnek: uçuş rotası
- **Yönlendirilmiş graf**
  - Bütün kenarları yönlendirilmiş graftır.
- **Yönlendirilmemiş graf**
  - Hiçbir kenarı yönlendirilmemiş graftır.

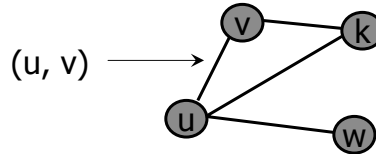
G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji)

### Komşu (Adjacent)

- Eğer  $(u, v) \in E$  ise  $u$  ve  $v$  node'ları komşudur.

$u$  ve  $v$  komşudur  
 $v$  ve  $w$  komşu değildir

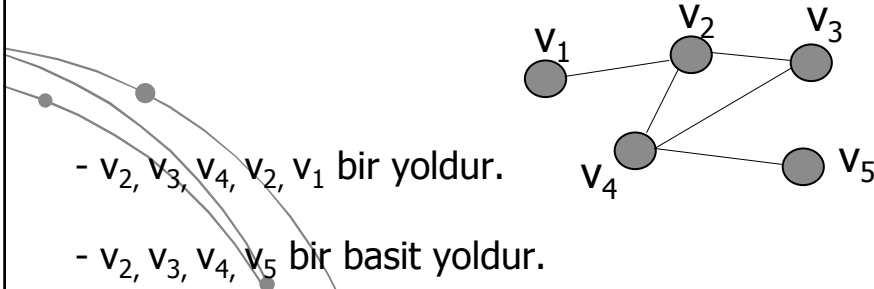


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji - devam)

### Yol ve basit yol

- Bir yol  $v_1$  den  $v_k$  ya kadar sıralı node'ları  $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)$  kenarlarıyla birbirine bağlar.
- Bir basit yolda her bir node sadece bir kez bulunur.

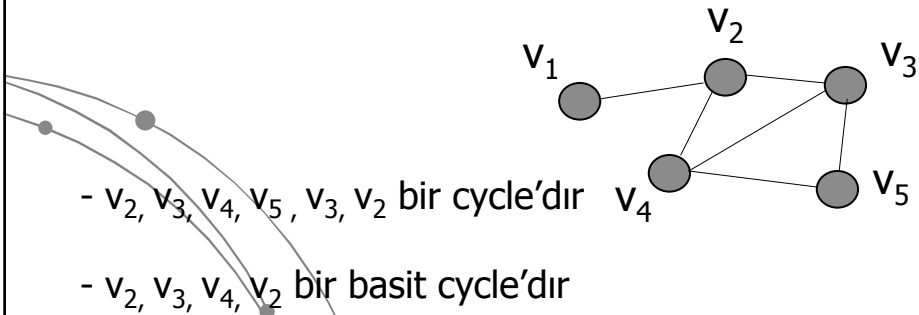


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji - devam)

### Cycle ve basit cycle

- Bir cycle bir yoldur ve başlama ve bitiş node'ları aynıdır.
- Bir basit cycle'da başlangıç ve bitiş node'ları hariç tüm node'lar sadece bir kez bulunur.

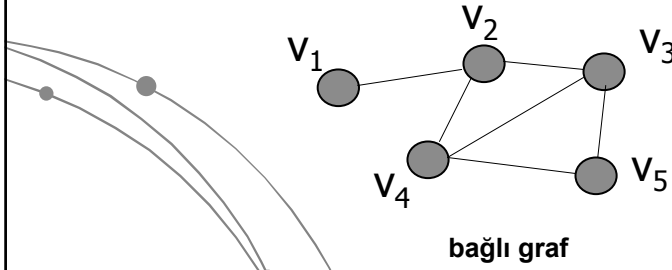


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji - devam)

### Bağlı ve bağlı olmayan graf

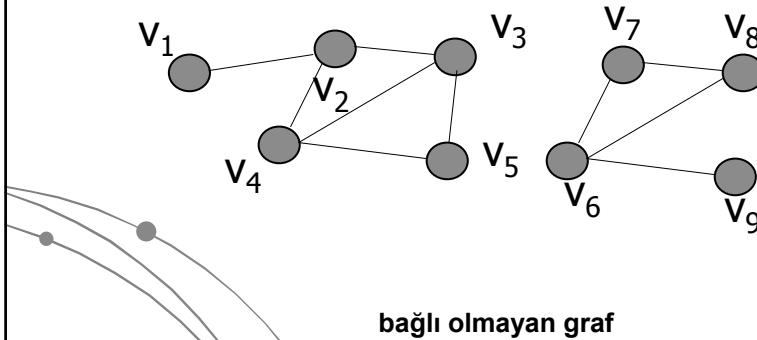
- Eğer bir graftaki tüm node'lar arasında en azından bir yol varsa **bağlı graf**tır.
- Eğer bir grafta herhangi iki node arasında yol bulunmuyorsa **bağlı olmayan graf**tır.



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji - devam)

### Bağlı ve bağlı olmayan graf (devam)

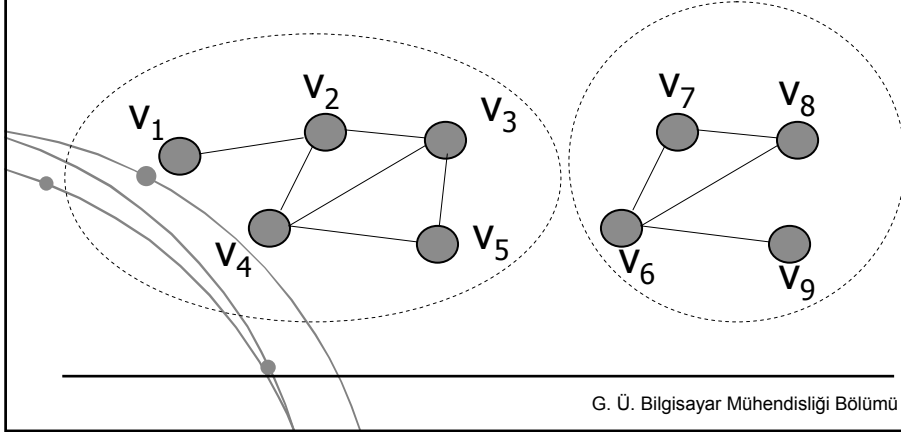


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Terminoloji - devam)

### Baęlı eleman (connected component)

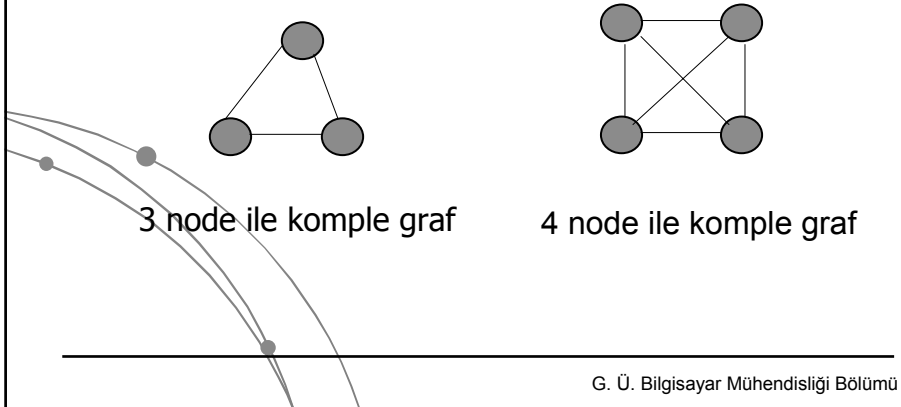
- Eęer bir graf baęlı deęilse, baęlı alt gruplara gore paralanabilir. Bu paraların herbirine baęlı eleman denir.



## Graflar (Terminoloji - devam)

### Komple graf

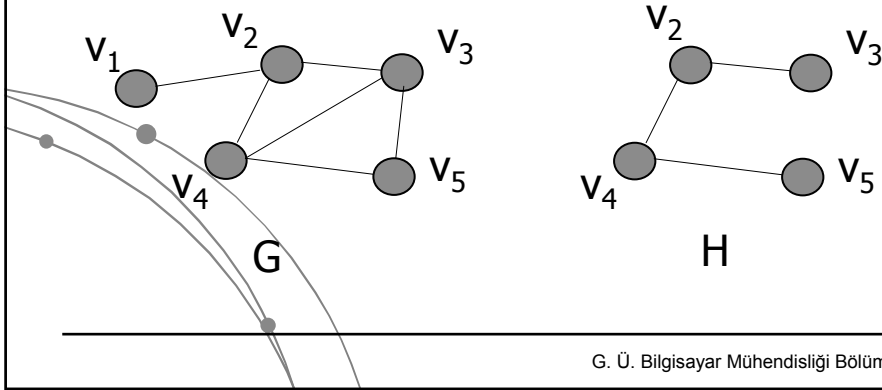
- Eęer bir graftaki her iki node arasında bir kenar varsa komple graftır.



## Graflar (Terminoloji - devam)

### Alt graf

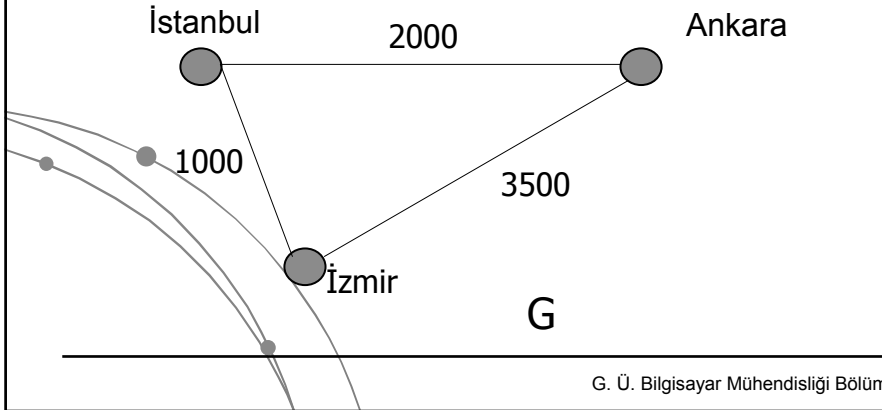
- $G(V, E)$  şeklinde gösterilen bir grafın alt grafı  $H(U, F)$  ise  $U \subseteq V$  ve  $F \subseteq E$  olur.



## Graflar (Terminoloji - devam)

### Ağırlıklandırılmış graf

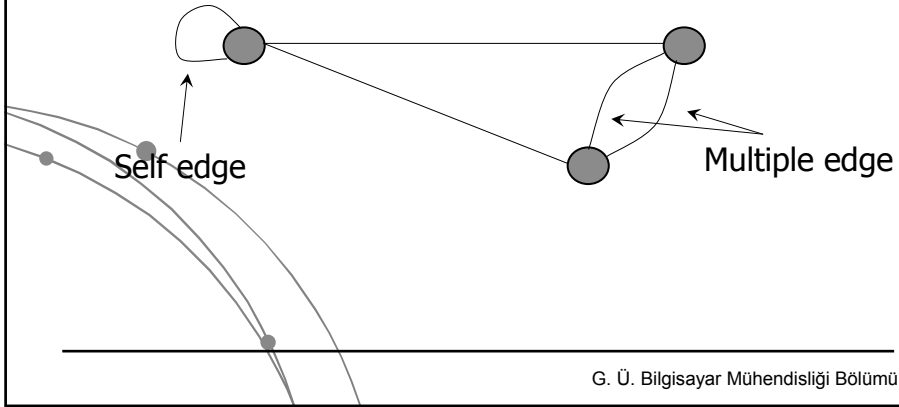
- Eğer  $G(V, E)$  şeklinde gösterilen bir grafta her  $E$  kenarına bir ağırlık değeri atanmış ise ağırlıklandırılmış graf olarak adlandırılır.



## Graflar (Terminoloji - devam)

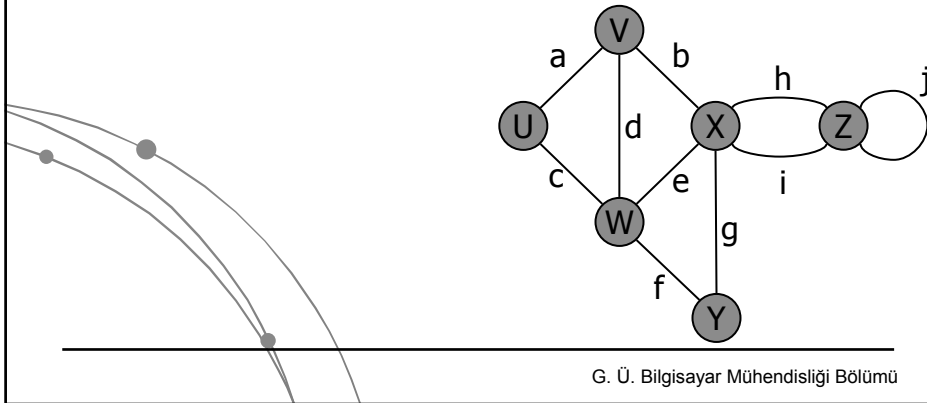
### Multigraf

- Multigraf iki node arasında birden fazla kenara sahip olan veya bir node'un kendi kendisini gösteren kenara sahip olan graftır.



## Graflar (Terminoloji - devam)

- a,b ve d kenarları V node'unun kenar bağlantılarıdır.
- X node'unun derecesi 5' tir.
- h ve i çoklu (multiple) kenarlardır.
- j kendi kendisine döngüdür (self loop).

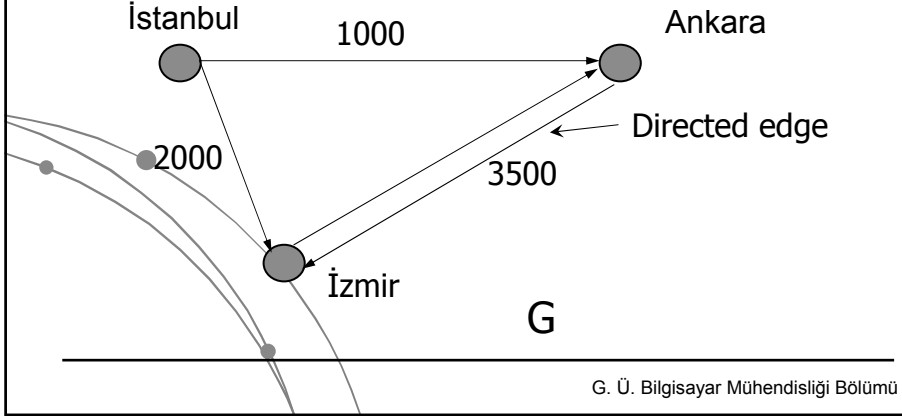




## Graflar (Terminoloji - devam)

### Yönlendirilmiş graf (Directed graph - Digraph)

- Eğer  $G (V, E)$  şeklinde gösterilen bir grafta her  $E$  kenarı bir yöne (directed edge) sahipse  $G$  yönlendirilmiş graftır.



## Graflar (Oluşturulması)

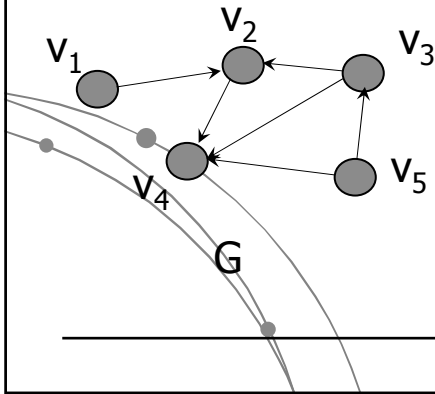
- Komşu matrisi (Adjacency matrix)  
Graf iki boyutlu matrisle gösterilir.
- Komşu listesi (Adjacency list)  
Graph  $n$  elemanlı  $m$  tane bağlı listeye gösterilir.  $n$  ilgili node'a komşu olan node sayısını,  $m$  ise toplam node sayısını ifade eder.

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Matrisi)

### • Yönlendirilmiş graf için komşu matrisi

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



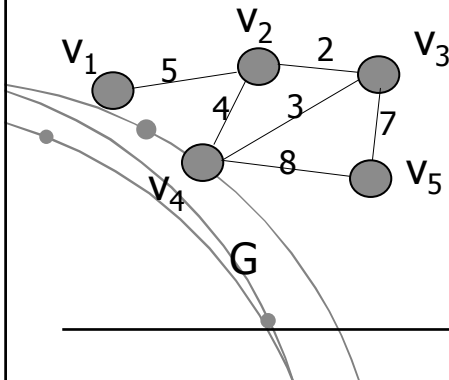
		1	2	3	4	5
		$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
1	$v_1$	0	1	0	0	0
2	$v_2$	0	0	0	1	0
3	$v_3$	0	1	0	1	0
4	$v_4$	0	0	0	0	0
5	$v_5$	0	0	1	1	0

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Matrisi)

### • Ağırlıklandırılmış ancak yönlendirilmemiş graf için komşu matrisi

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

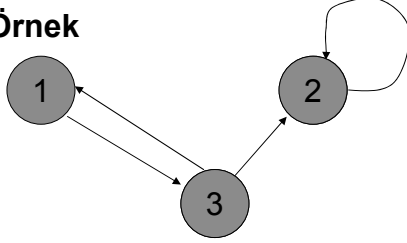


		1	2	3	4	5
		$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
1	$v_1$	$\infty$	5	$\infty$	$\infty$	$\infty$
2	$v_2$	5	$\infty$	2	4	$\infty$
3	$v_3$	$\infty$	2	$\infty$	3	7
4	$v_4$	$\infty$	4	3	$\infty$	8
5	$v_5$	$\infty$	$\infty$	7	8	$\infty$

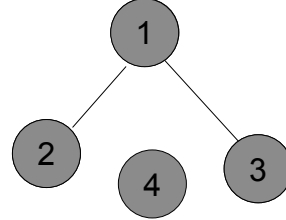
G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Matrisi)

Örnek



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

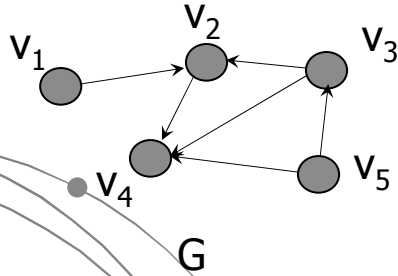


	1	2	3	4
1	0	1	1	0
2	1	0	0	0
3	1	0	0	0
4	0	0	0	0

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Listesi)

• Yönlendirilmiş graf için komşu listesi

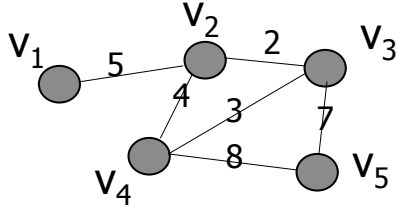


1	$v_1 \rightarrow v_2$
2	$v_2 \rightarrow v_4$
3	$v_3 \rightarrow v_2 \rightarrow v_4$
4	$v_4$
5	$v_5 \rightarrow v_3 \rightarrow v_4$

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Listesi)

- Yönlendirilmiş ve ağırlıklandırılmış graf için komşu listesi

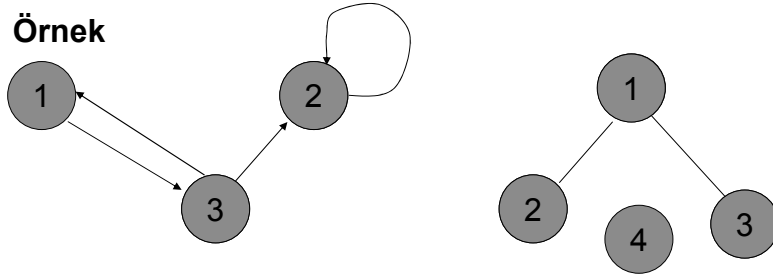


1	$v_1$	$\rightarrow v_2(5)$
2	$v_2$	$\rightarrow v_1(5) \rightarrow v_3(2) \rightarrow v_4(4)$
3	$v_3$	$\rightarrow v_2(2) \rightarrow v_4(3) \rightarrow v_5(7)$
4	$v_4$	$\rightarrow v_2(4) \rightarrow v_3(3) \rightarrow v_5(8)$
5	$v_5$	$\rightarrow v_3(7) \rightarrow v_4(8)$

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Oluşturulması-Komşu Listesi)

Örnek



1	$\rightarrow 3$
2	$\rightarrow 2$
3	$\rightarrow 1 \rightarrow 2$

1	$\rightarrow 2 \rightarrow 3$
2	$\rightarrow 1$
3	$\rightarrow 1$
4	$\rightarrow$

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Komşu Matrisi-Komşu Listesi)

### • Avantajları dezavantajları

#### Komşu matrisi

- Çok fazla alana ihtiyaç duyar.
- Daha az hafızaya ihtiyaç duyulması için sparse matris tekniklerinin kullanılması gerekir.
- Herhangi iki node'un komşu olup olmadığına çok kısa sürede karar verilebilir.

#### Komşu listesi

- Bir node'un tüm komşularına hızlı bir şekilde ulaşılır.
- Daha az alana ihtiyaç duyar.
- Oluşturulması matrise göre daha zor olabilir.

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

### • Depth first dolaşma

- Bir v node'una gidildikten sonra v node'unun bir komşusu seçilir ve ziyaret edilir. Ardından onun bir komşusu seçilir ve ard arda komşu seçimi yapılarak devam edilir. Komşu kalmadığında geri dönülür.

### • Breadth first dolaşma

- Bir v node'una gidildikten sonra v node'unun sırasıyla tüm komşu node'larına gidilir ardından tüm komşu node'ların komşu node'larına gidilir.

G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolařma - Traversal)

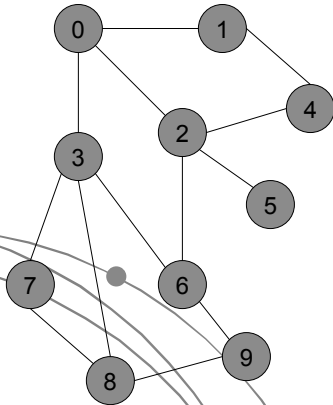
### • Depth first arama iřlem adımları

1. Önce bir bařlangıç node'u seçilir ve ziyaret edilir.
2. Seçilen node'un bir komřusu seçilir ve ziyaret edilir.
3. 2.adım ziyaret edecek komřu kalmayınca kadar tekrar edilir.
4. Komřu kalmadıđında tekrar geri dönölür ve önceki ziyaret edilmiř node'lar için adım 2 ve 3 tekrar edilir.

G. Ü. Bilgisayar Mühendisliđi Bölümü

## Graflar (Dolařma - Traversal)

### • Depth first arama

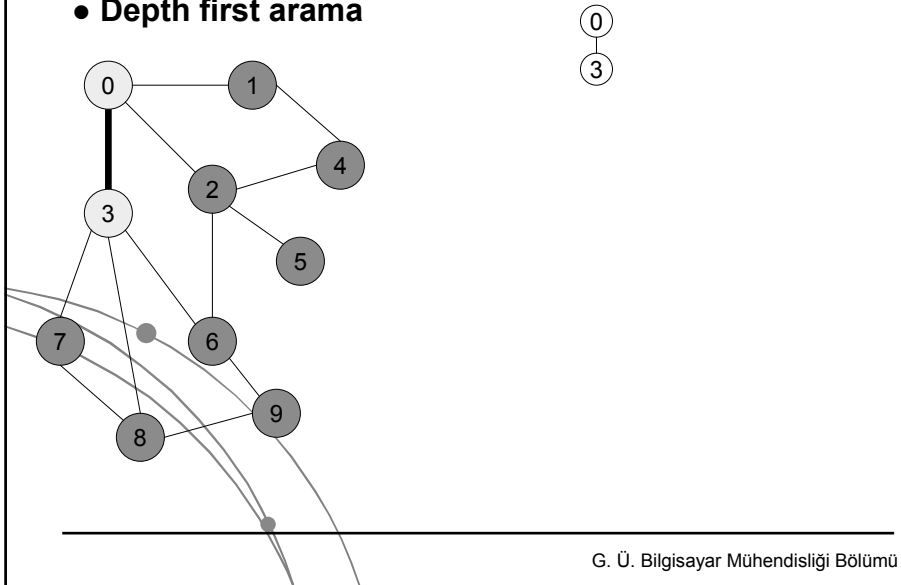


1. `s` node'unu seç
2. `visit s`  
`// örn. ekrana yaz`
3. `for each edge <s, U>`  
`// U komřu node`
4. `if U is not visit`
5. `DFS(G, U)`

G. Ü. Bilgisayar Mühendisliđi Bölümü

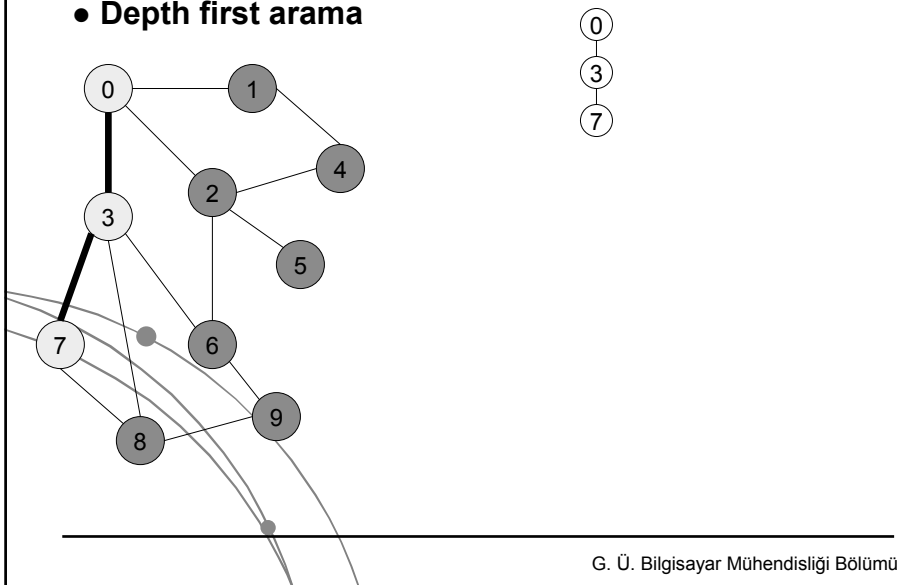
## Graflar (Dolaşma - Traversal)

### • Depth first arama



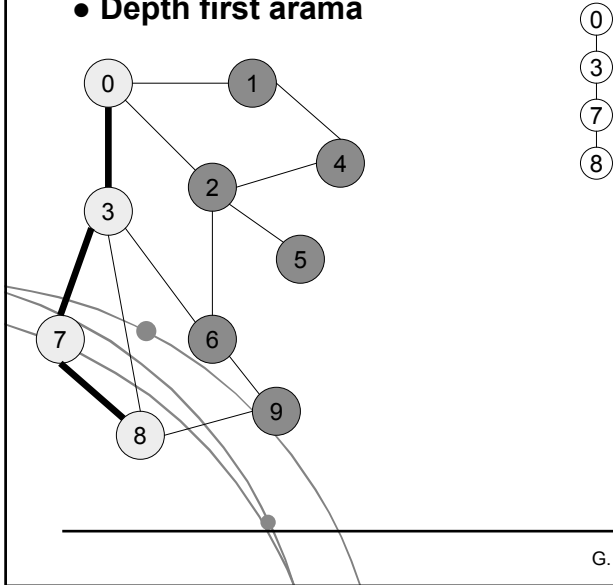
## Graflar (Dolaşma - Traversal)

### • Depth first arama



## Graflar (Dolaşma - Traversal)

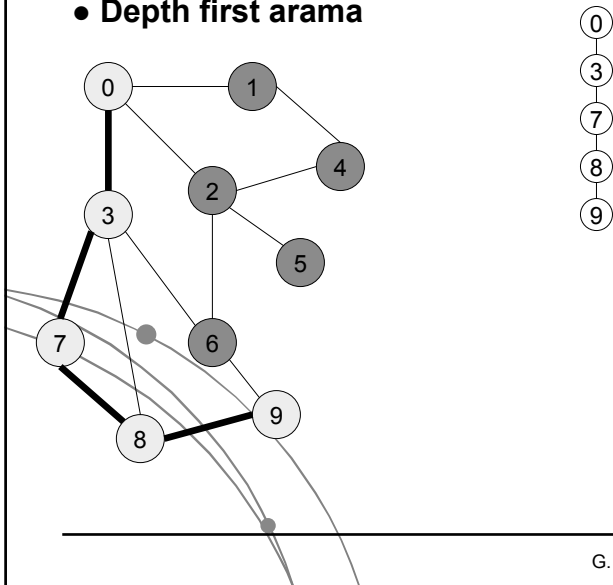
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

### • Depth first arama

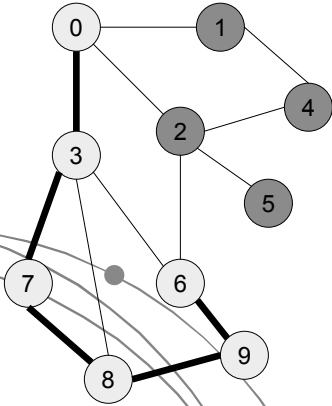


G. Ü. Bilgisayar Mühendisliği Bölümü



## Graflar (Dolaşma - Traversal)

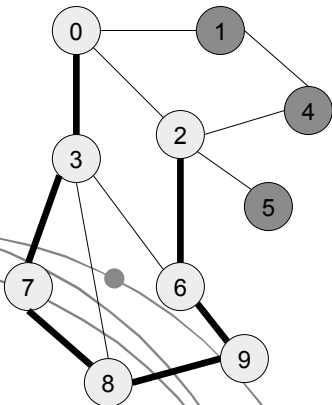
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

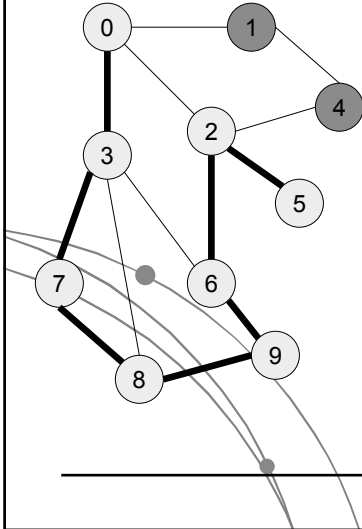
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

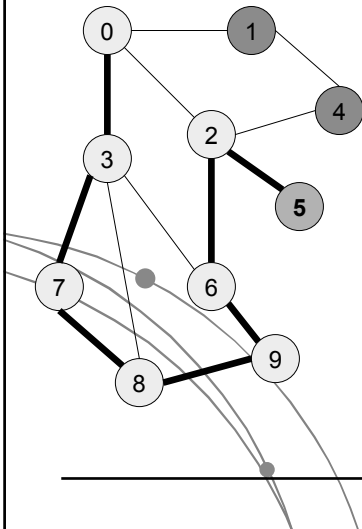
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

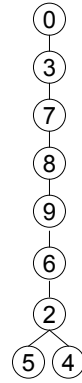
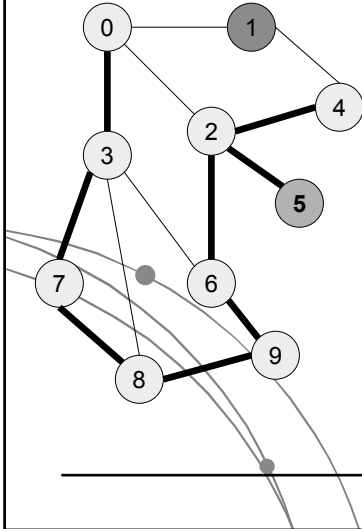
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

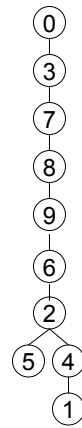
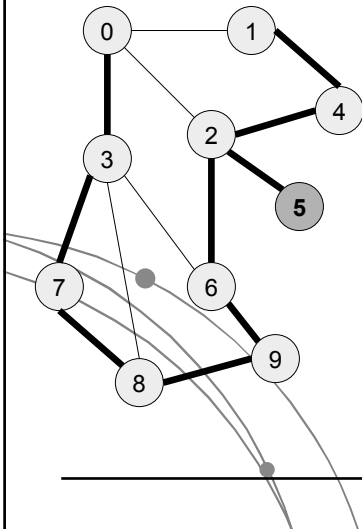
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

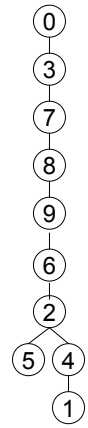
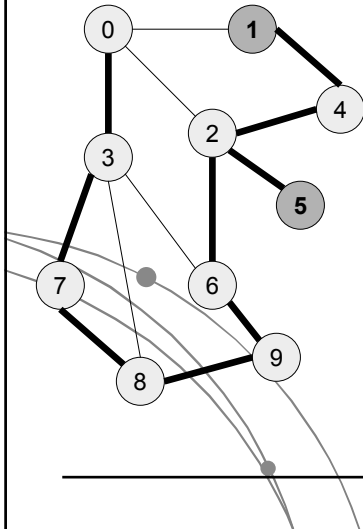
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

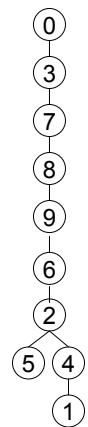
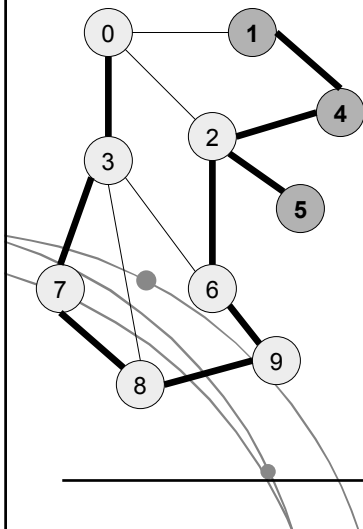
- Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

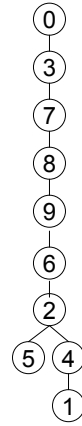
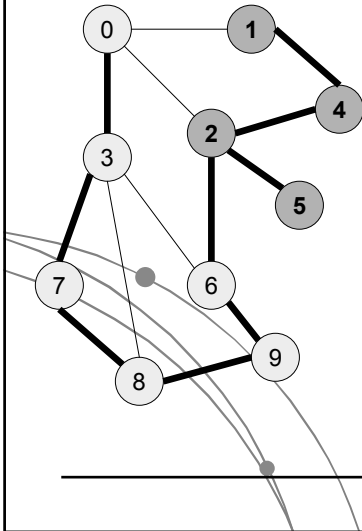
- Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

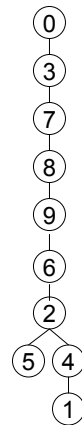
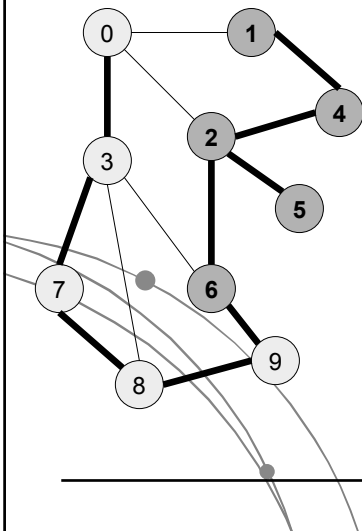
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

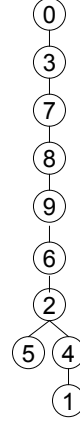
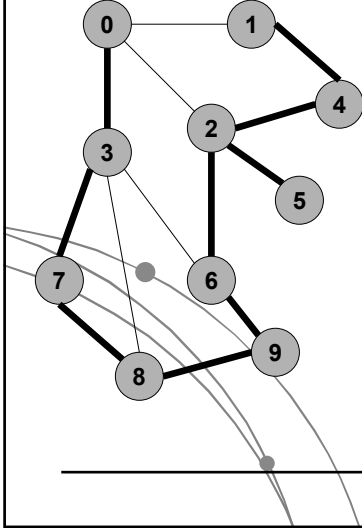
### • Depth first arama



G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

### • Depth first arama

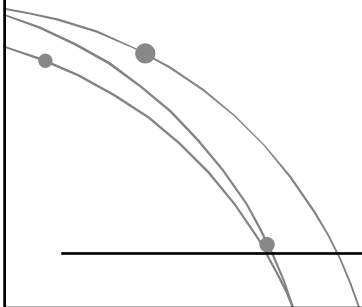


G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar (Dolaşma - Traversal)

### • Breadth first arama işlem adımları

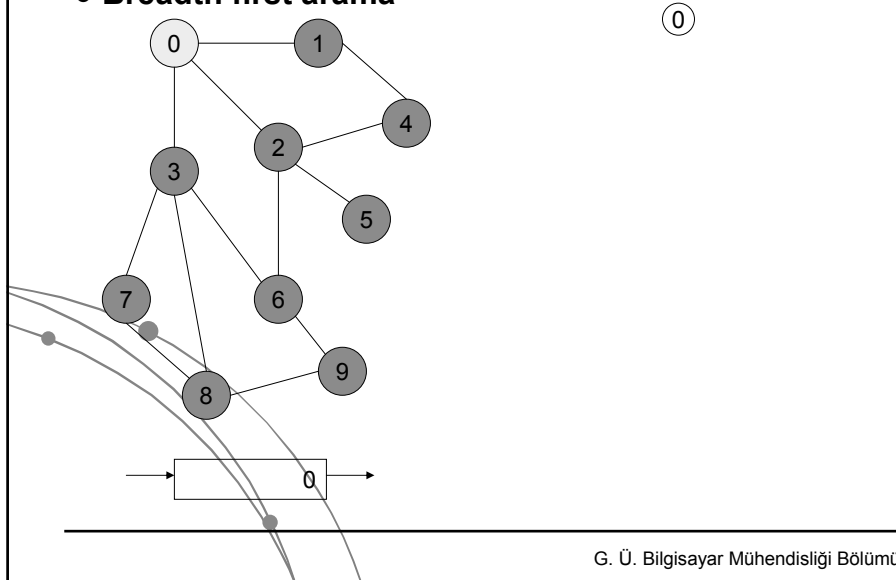
1. Breadth first arama ağaçlardaki level order aramaya benzer.
2. Seçilen node'un tüm komşuları sırayla seçilir ve ziyaret edilir.
3. Her komşu queue içerisine atılır.
4. Komşu kalmadığında Queue içerisindeki ilk node alınır ve 2.adıma gidilir.



G. Ü. Bilgisayar Mühendisliği Bölümü

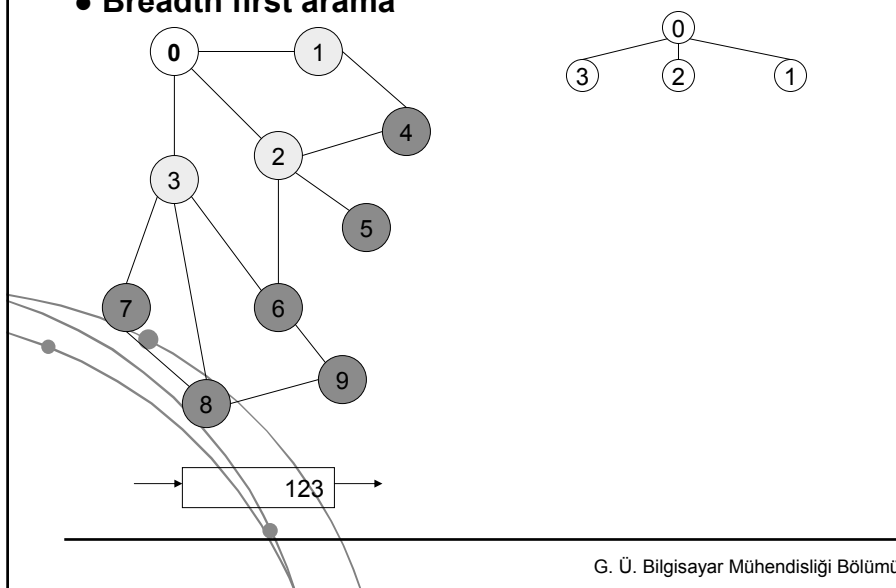
## Graflar (Dolaşma - Traversal)

### • Breadth first arama



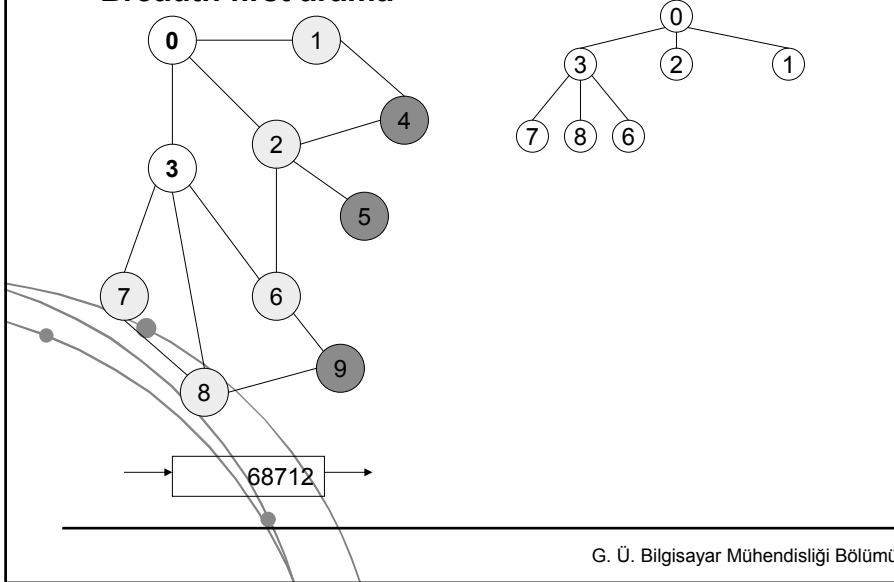
## Graflar (Dolaşma - Traversal)

### • Breadth first arama



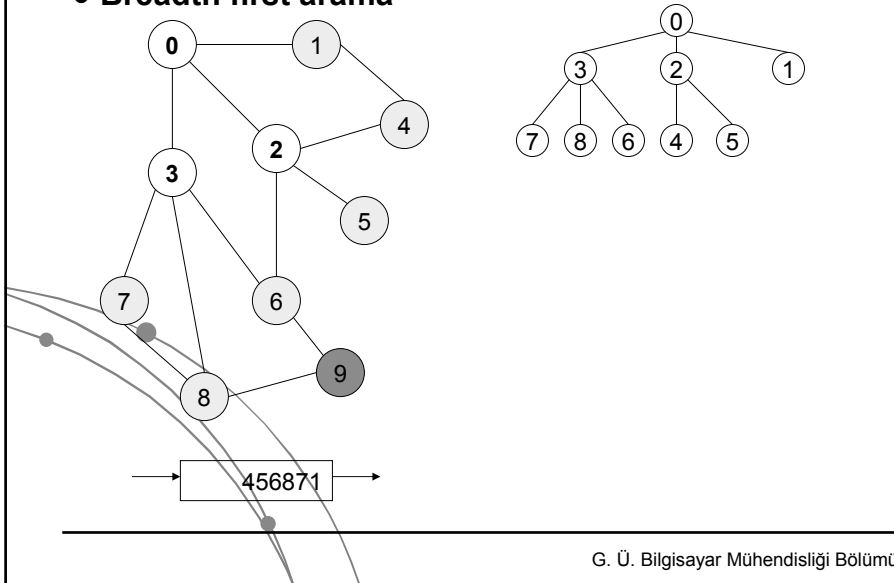
## Graflar (Dolaşma - Traversal)

- Breadth first arama



## Graflar (Dolaşma - Traversal)

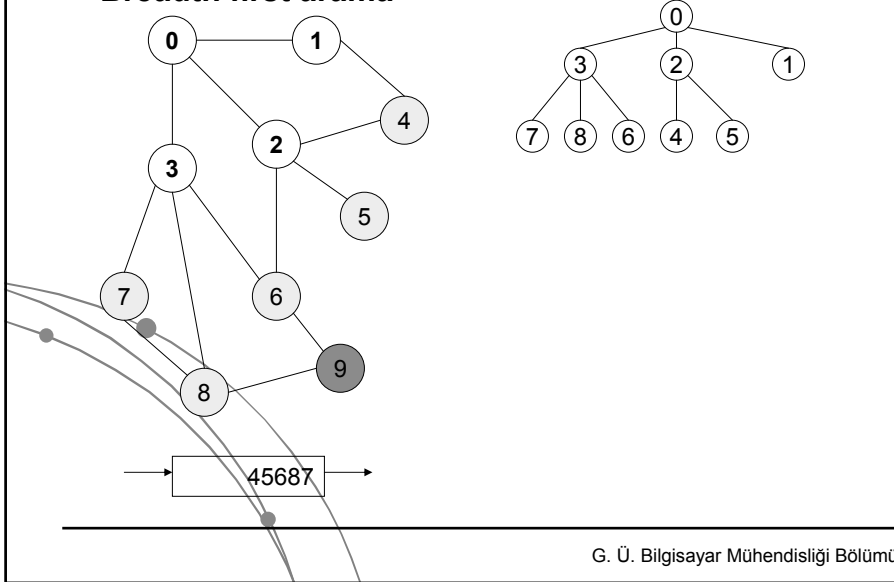
- Breadth first arama





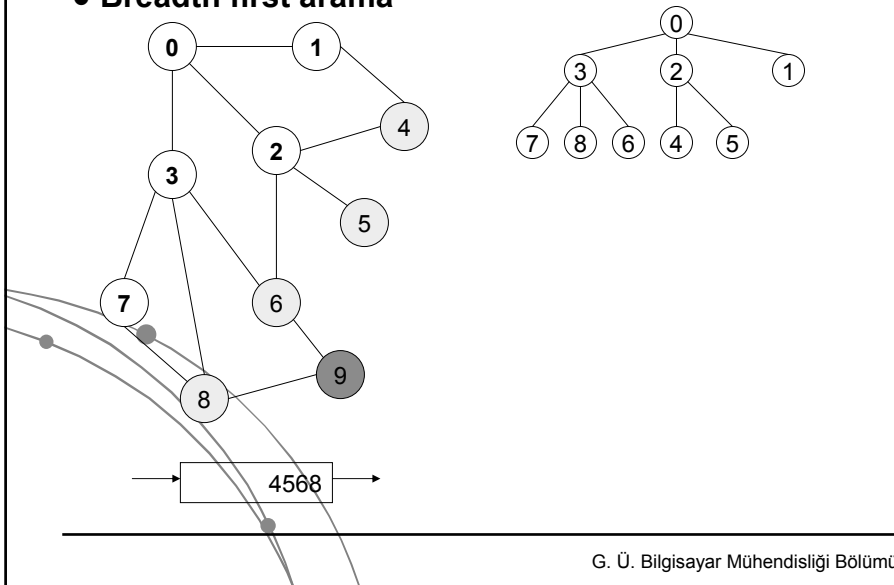
## Graflar (Dolaşma - Traversal)

- Breadth first arama



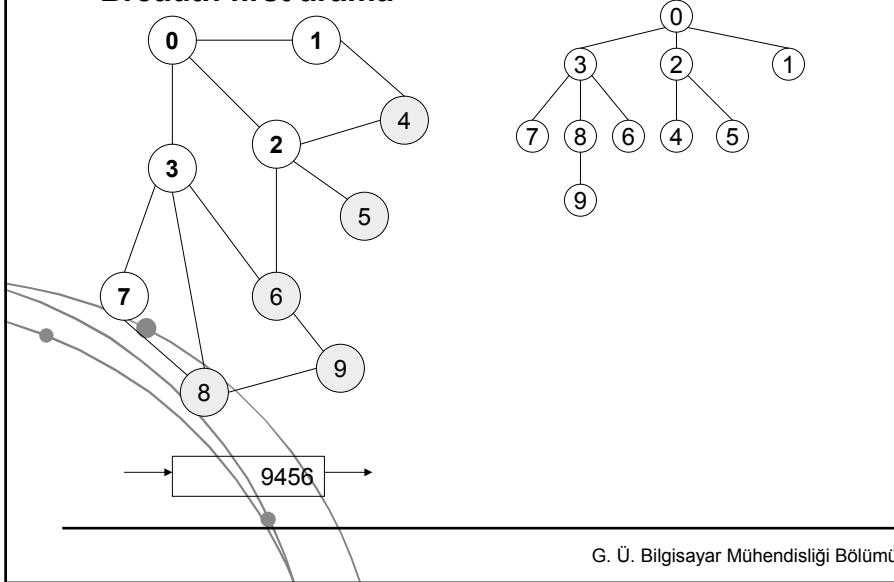
## Graflar (Dolaşma - Traversal)

- Breadth first arama



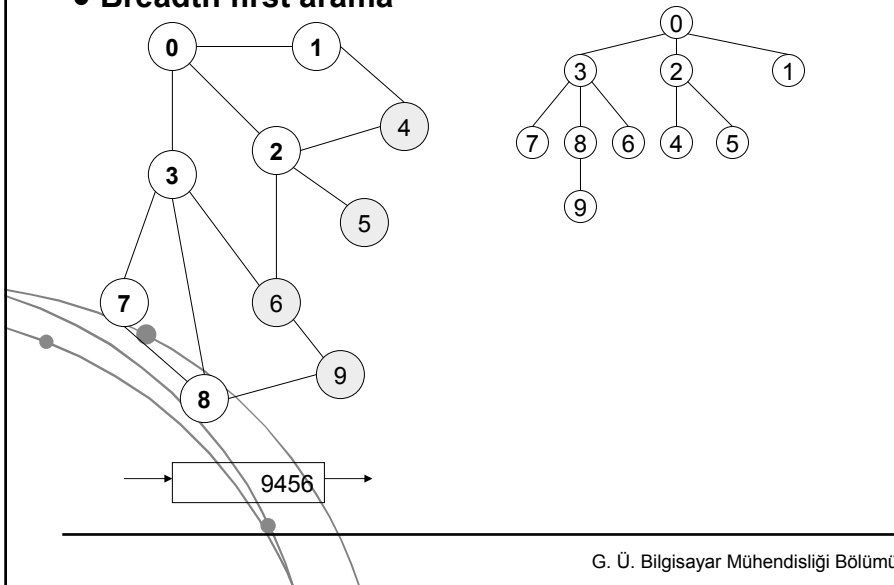
## Graflar (Dolaşma - Traversal)

- Breadth first arama



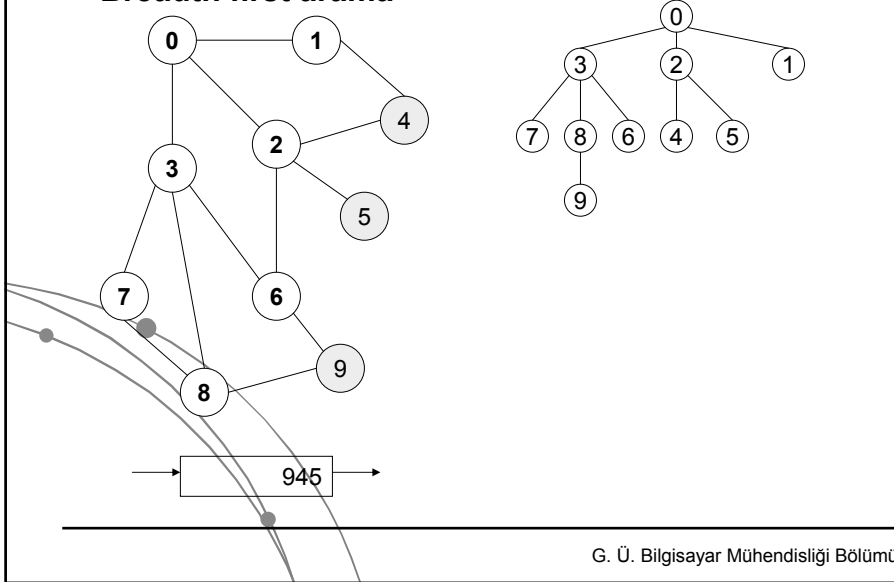
## Graflar (Dolaşma - Traversal)

- Breadth first arama



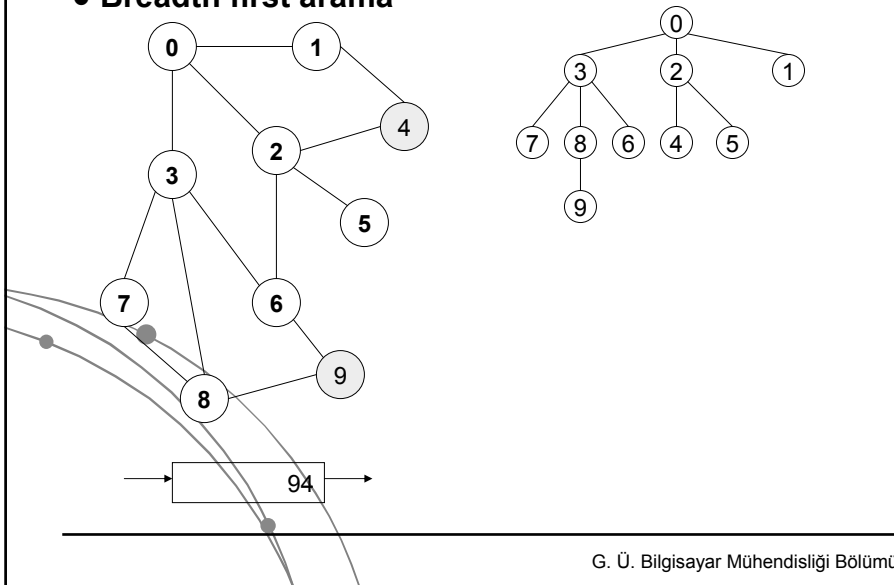
## Graflar (Dolaşma - Traversal)

### • Breadth first arama



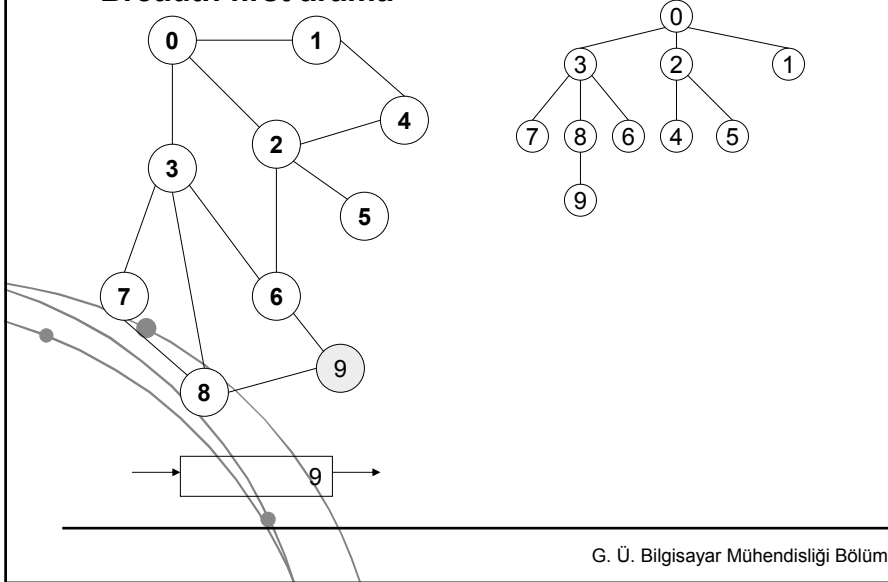
## Graflar (Dolaşma - Traversal)

### • Breadth first arama



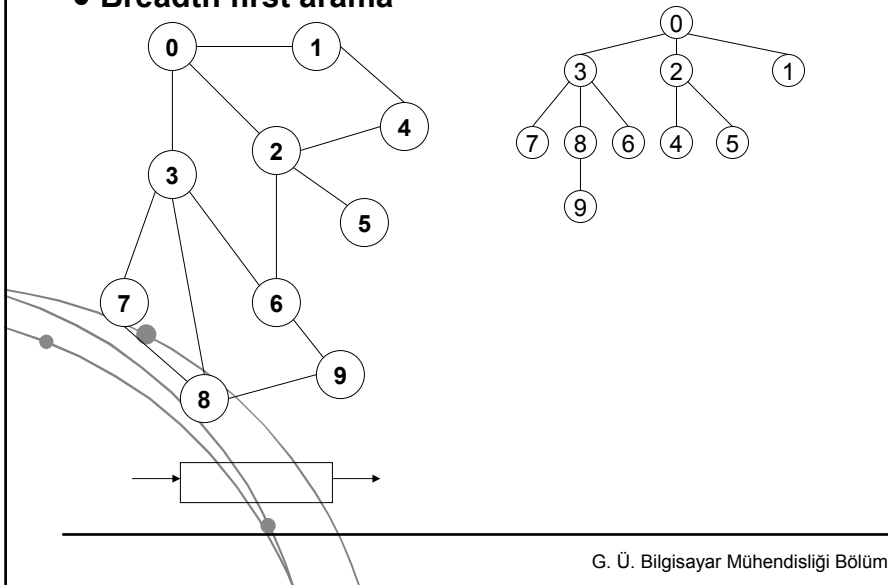
## Graflar (Dolaşma - Traversal)

### • Breadth first arama



## Graflar (Dolaşma - Traversal)

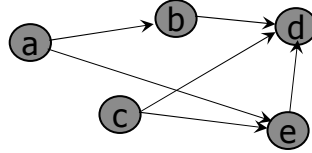
### • Breadth first arama



## Graflar (Topolojik sıralama)

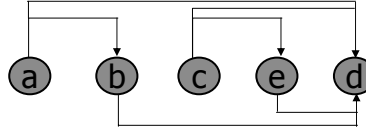
### • Topolojik sıralama

- Topolojik sıralama bir graftaki tüm node'ların doğrusal sıralamasıdır.
- Node'lar arasında öncelik sırası gözönüne alınır.



- Örnekteki grafta topolojik sıralama aşağıdaki gibi yapılabilir.

- 1- a, c, b, e, d
- 2- c, a, b, e, d



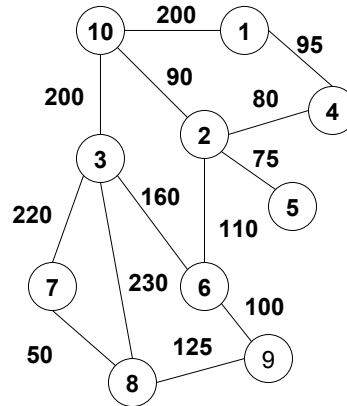
G. Ü. Bilgisayar Mühendisliği Bölümü

## Graflar

### Haftalık Ödev:

- 10 tane şehir için bir graf yapısı oluşturunuz. Her şehirden komşu şehirlere olan uzaklık kenar ağırlıkları olarak kullanılacaktır.

- Herhangi bir şehirden başlayarak tüm şehirleri dolaşmak için gerekli olan algoritmayı depth first dolaşmayla yapınız.



G. Ü. Bilgisayar Mühendisliği Bölümü