

6.3.2 komşuluk listesi

- seyrek bağlantılı durumlar için daha verimli
- giriş bilgisini link list ile tut (zincirleme mekanizması)
- aşağıdaki resim için,

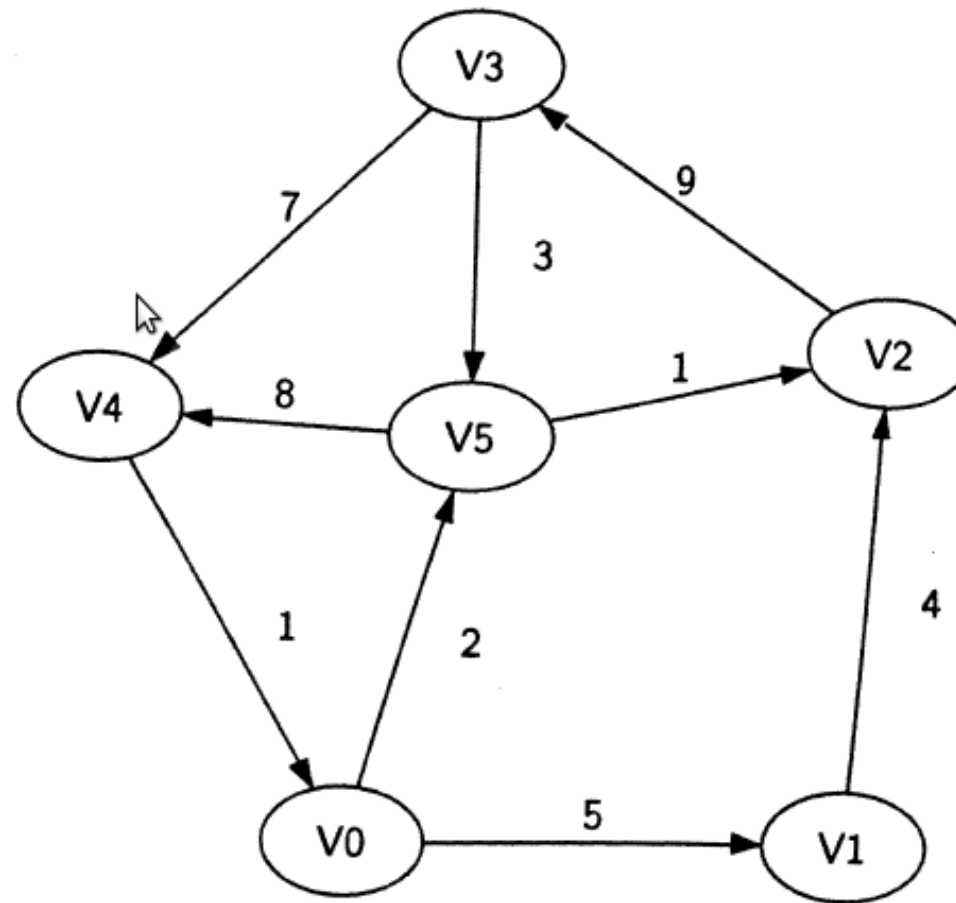
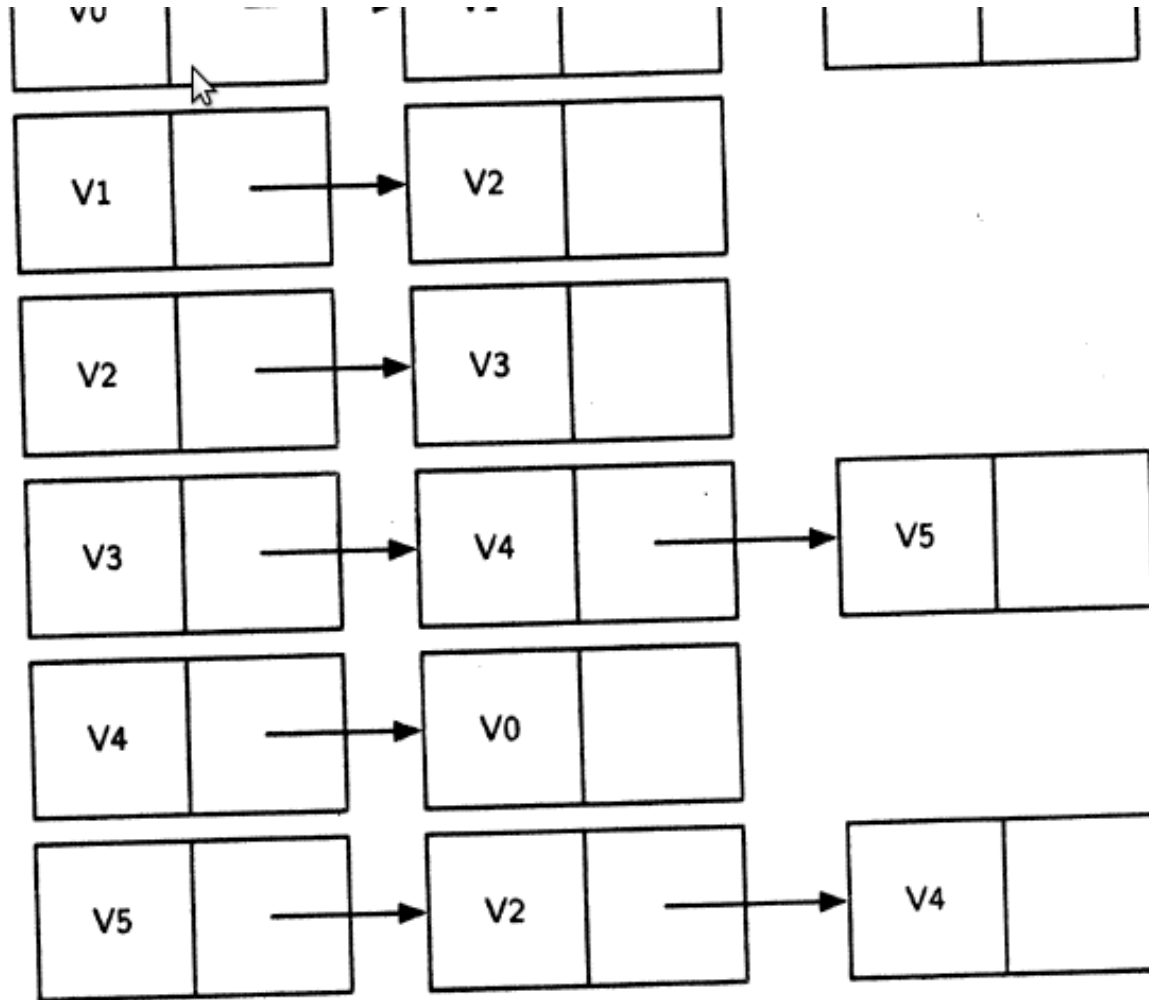


Figure 6.2: A Simple Example of a Directed Graph

komşuluk listesi

komşuluk listesi



→ satırlar: tüm düğümler

→ sütunlar: oluşturduğu kırışlar

6.3.3 gerçekte

- komşuluk listesi, sözlükler ve listelerle gerçekleştirilebilir
- satırlar: düğümler: sözlük veri yapısıyla temsil edilir
- sözlük: key: düğüm kimliği
- sözlük: value: Vertex nesnesi
- her bir Vertex, bağlı olduğu diğer düğümleri
- oluşturduğu girişin diğer ucundaki düğümleri tutmakla sorumlu
- çizge SVT için iki sınıf: Graph, Vertex

demo

bittiğinde böyle kullanacağız

```
1  >>> g = Graph()
2  >>> for i in range(6):
3  ...     g.addVertex(i)
4  ...
5  >>> g.vertList
6  ...
7  >>> g.addEdge(0, 1)
8  >>> g.addEdge(0, 5)
9  ...
10 >>> for v in g:
11 ...     for w in v.getAdj():
12 ...         print "(", v, ",", w, ")"
13 ...
14 ( 0 , 1 )
15 ( 0 , 5 )
16 ...
```

gerçekleme

Vertex class

```
1      class Vertex:
2          def __init__(self,num):
3              self.id = num
4              self.adj = []
5              self.color = 'white'
6              self.dist = sys.maxint
7              self.pred = None
8              self.disc = 0
9              self.fin = 0
10             self.cost = {}
11
12         def addNeighbor(self,nbr,cost=0):
13             self.adj.append(nbr)
14             self.cost[nbr] = cost
15
16         def __str__(self):
17             return str(self.id) + ":color " + self.color + \
18                 ":dist " + str(self.dist) +
19                 ":pred [" + str(self.pred)+ "]\n"
```

gerçekleme

Vertex class

```
1      def getCost(self,nbr):
2          return self.cost[nbr]
3      def setCost(self,nbr,cost):
4          self.cost[nbr] = cost
5      def setColor(self,color):
6          self.color = color
7      def setDistance(self,d):
8          self.dist = d
9      def setPred(self,p):
10         self.pred = p
11      def setDiscovery(self,dttime):
12         self.disc = dttime
13      def setFinish(self,ftime):
14         self.fin = ftime
15      def getFinish(self):
16         return self.fin
17      def getDiscovery(self):
18         return self.disc
```

gerçekleme

Vertex class

```
1      def getPred(self):  
2          return self.pred  
3      def getDistance(self):  
4          return self.dist  
5      def getColor(self):  
6          return self.color  
7      def getAdj(self):  
8          return self.adj  
9      def getId(self):  
10         return self.id
```

açıklama

s31:

→ `__init__`: self.adj: liste

→ self.cost: sözlük

gerçekleme

Graph class

```
1      class Graph:
2          def __init__(self):
3              self.vertList = {}
4              self.numVertices = 0
5
6          def addVertex(self, key):
7              self.numVertices = self.numVertices + 1
8              newVertex = Vertex(key)
9              self.vertList[key] = newVertex
10             return newVertex
11
12         def getVertex(self, n):
13             if self.vertList.has_key(n):
14                 return self.vertList[n]
15             else:
16                 return None
```

gerçekleme

Graph class

```
1      def has_key(self,n):
2          return self.vertList.has_key(n)
3
4      def addEdge(self,f,t,c=0):
5          if not self.vertList.has_key(f):
6              nv = self.addVertex(f)
7          if not self.vertList.has_key(t):
8              nv = self.addVertex(t)
9          self.vertList[f].addNeighbor(self.vertList[t],c)
10
11     def getVertices(self):
12         return self.vertList.values()
13
14     def __iter__(self):
15         return self.vertList.itervalues()
```

açıklama

s31:

- `__init__`: `self.vertList`: sözlük
- `addVertex`: `newVertex = Vertex(key)`: `Vertex` sınıfının yapıcısı
- `getVertex`: `self.vertList`, sözlük olduğuna göre `"has_key(n)"` anlamlı
- `self`, `n` ile bağlıysa düğümü, değilse `None` döndür
- `addEdge`: `f`'dan `t`'ya bir giriş ekle (s32)
- `f` ve `t` yoksa oluştur
- `getVertices`: `self.vertList` (`Graph` class, satır9, sözlük) o yüzden `values()` anlamlı
- `__iter__`: `self.vertList`, sözlük olduğundan, sözlükler için `itervalues` anlamlı
- bununla `for v in g` anlamlı
- ayrıca `for w in v.getAdj`'deki `getAdj`, liste döndürür (komşuluk listesi)