数据结构

图

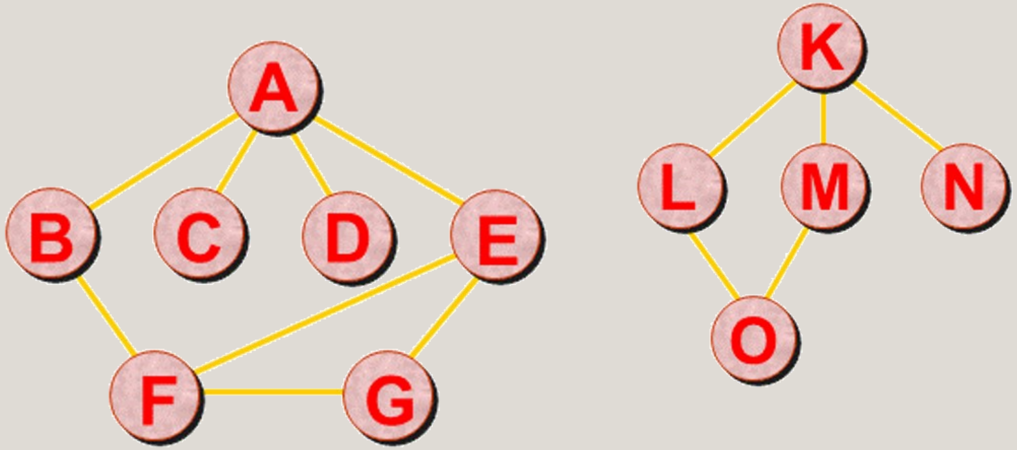
2020年11月

深圳大学电子与信息工程学院 周飞

第六章 图

第四节 图的连通性

- 一、无向图的连通性
  - 如果无向图中，存在不连通的顶点，则称之为非连通图。

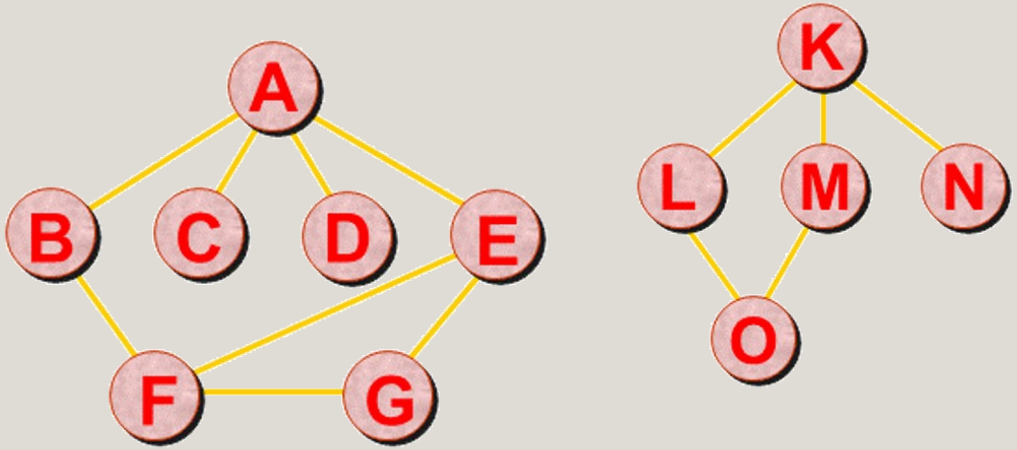


这是一个非连通图，其包括有12个顶点。

第六章 图

第四节 图的连通性

- 二、无向图的连通分量
  - 非连通图的极大连通子图叫做连通分量。

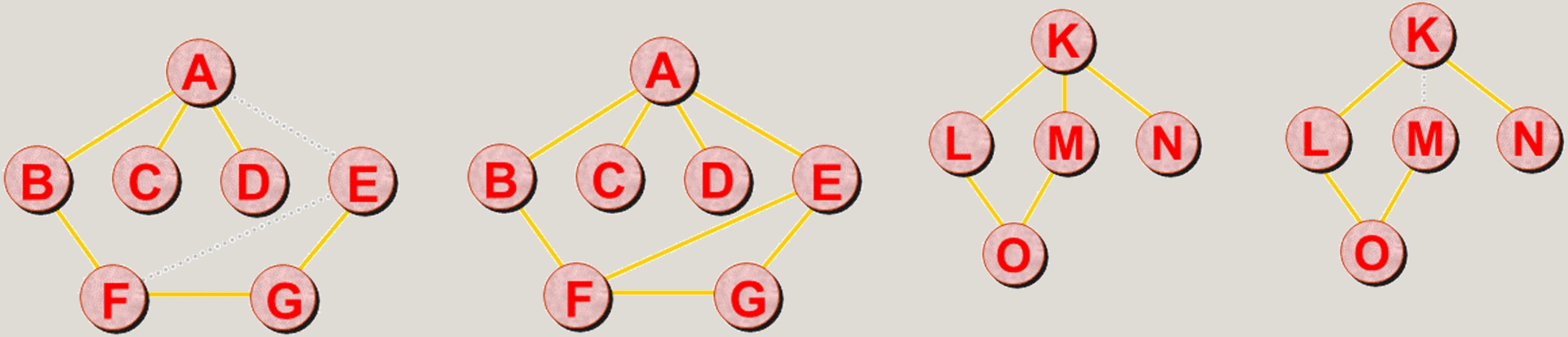


这是一个非连通图，其包括有两个连通分量。

第六章 图

第四节 图的连通性

- 三、无向图的生成树
  - 如果从非连通图的每一个连通分量中的一个顶点出发进行DSF遍历：



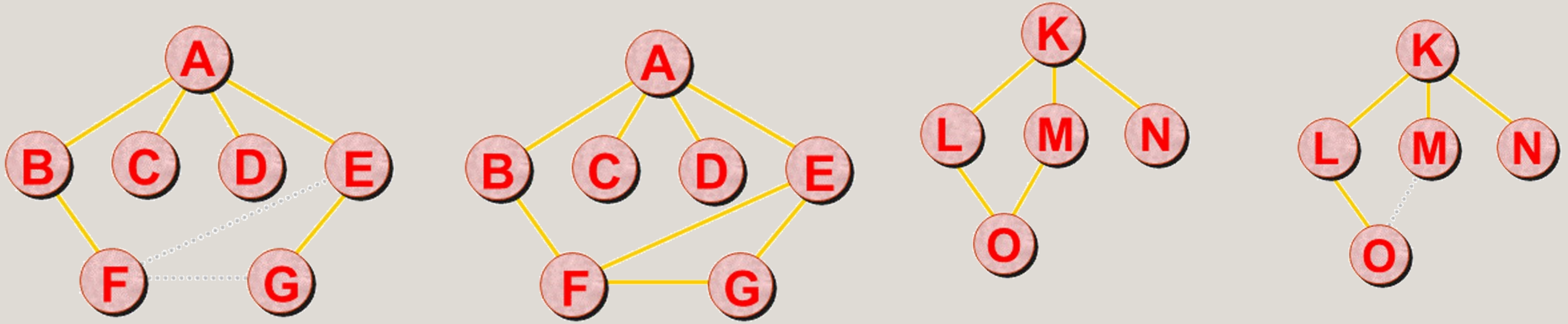
  - 可以获得无向图中每个连通分量的生成树，称为DSF生成树。

所有连通分量的生成树构成了非连通图的生成森林。

第六章 图

第四节 图的连通性

- 三、无向图的生成树
  - 如果从非连通图的每一个连通分量中的一个顶点出发进行BSF遍历：



  - 也可以获得无向图中每个连通分量的生成树，称为BSF生成树。

第六章 图

第四节 图的连通性

- 四、最小生成树
  - 如果无向图中，边上有权值，则称该无向图为无向网
  - 如果无向网中每个顶点都相通，称为连通网
  - 最小生成树（Minimum Cost Spanning Tree）是代价最小的连通网的生成树，即该生成树上的边的权值和最小。

第六章 图

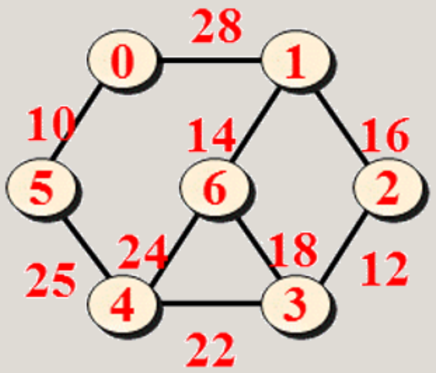
第四节 图的连通性

- 四、最小生成树（准则）
  - 使用n-1条边连接网络中的n个顶点；
  - 不能使用产生回路的边；
  - 所有的顶点要被连接到；
  - 各边上权值的总和达到最小。

第六章 图

第四节 图的连通性

- 四、最小生成树
  - 思路？



U表示已经被连接的顶点的集合

V-U表示未被连接的顶点

假设N=(V, E)是连通网

TE是N上最小生成树中边的集合

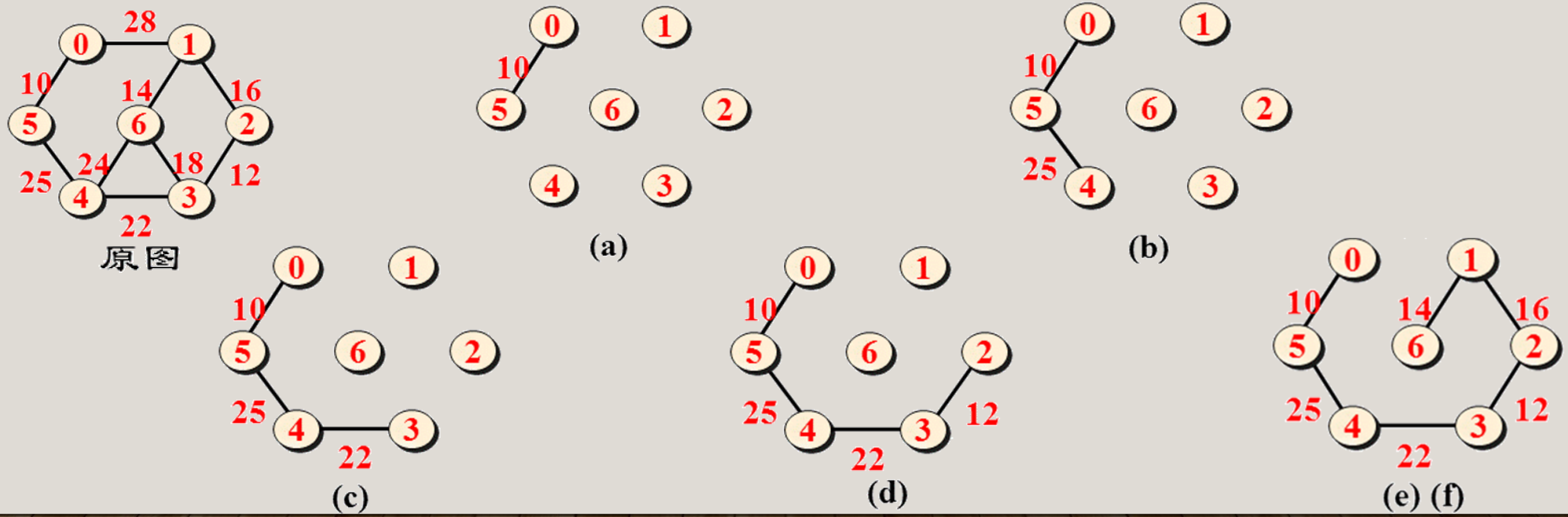
1. TE={}，U={u_0}，（u_0∈V）
2. 在所有u∈U, v∈V-U的边(u, v)∈E中找一条代价最小的边(u, v_0)并入集合TE，同时v_0并入U
3. 重复2，直到U=V

普里姆（Prim）算法

第六章 图

第四节 图的连通性

- 四、最小生成树

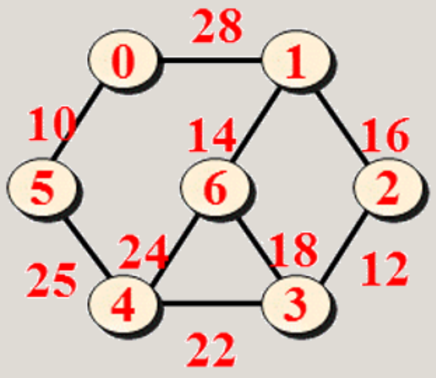


原图 (a) (b) (c) (d) (e) (f)

第六章 图

第四节 图的连通性

- 四、最小生成树
  - 其他思路？



假设N=(V, E)是连通网

在普里姆算法中，从一个顶点出发，依次把代价最小的边加入生成树。

可不可以从一个边出发，依次把顶点连接到生成树？

1. 初始状态是什么？

   (a)没有边：每个顶点自成一个连通分量，即非连通图T=（V,{}）

2. 选择哪条边？

   (b-1)选择的是代价最小的边

3. 加入边时，怎么防止出现回路？

   (b-2)选择的边要满足其依附的两个顶点属于不同的连通分量

(c)满足(b-1)和(b-2)的边加入T中

4. 怎么保证全部的顶点都连接上了？

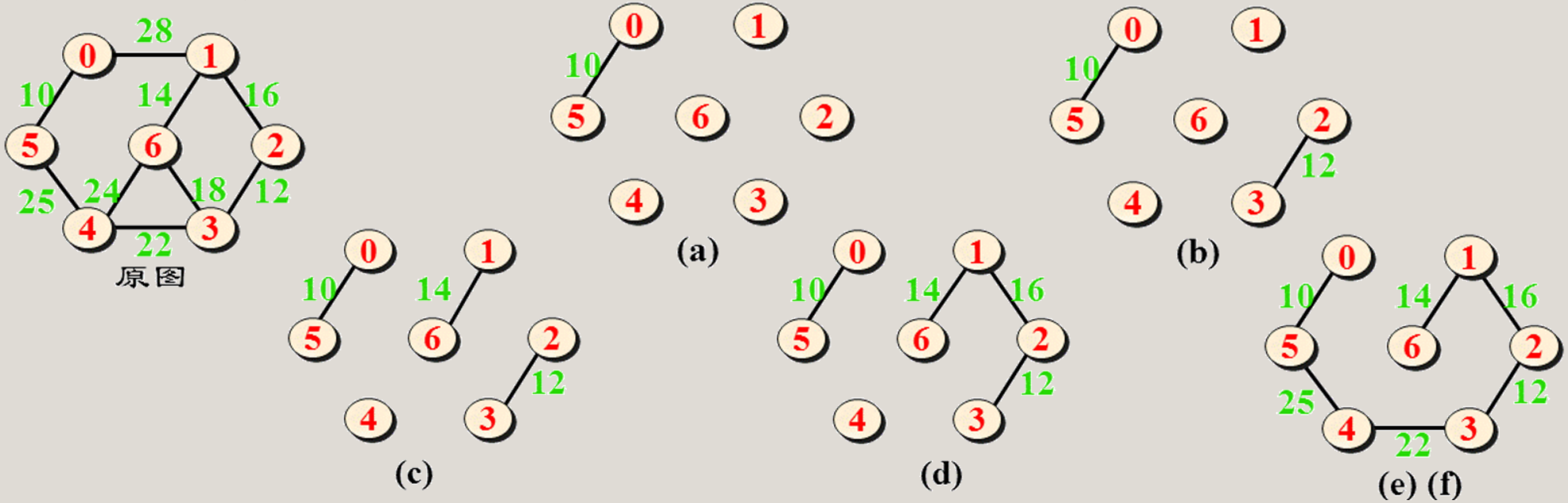
   (d)重复(c)直到T中所有的顶点都在一个连通分量上

克鲁斯卡尔(Kruskal)算法

第六章 图

第四节 图的连通性

- 四、最小生成树



第六章 图

第五节 最短路径

- 一、最短路径
  - 最短路径是求从图（或网）中某一顶点到其余各顶点的路径长度最短的路径。（如果有n个顶点，对于给定的顶点有最短路径有n-1条）

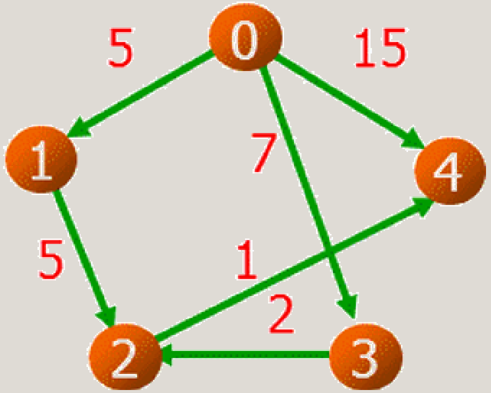
最短路径与最小生成树主要有三点不同：

1. **最短路径的操作对象主要是有向图(网)，而最小生成树的操作对象是无向图(网)**
2. **最短路径有一个始点，最小生成树没有**
3. **最短路径关心的是始点到每个顶点的路径最短，而最小生成树关心的是整个树的代价最小**

第六章 图

第五节 最短路径

- 二、算法
  - 思路？



如果0是始点v_0

从0出发的路：

有三条，到1的代价是5　　　到2没有路径，其代价是∞

到3的代价是7

到4的代价是15

把当前代价可以记录到一个辅助数组D中，D=[5, ∞,7,15]。

可以得到什么结论？ 0到1的最短路径就是<0,1>

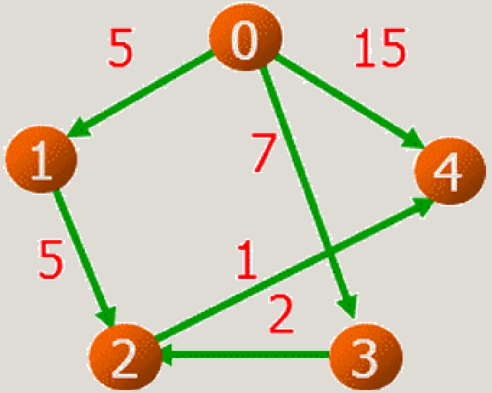
可以确定0到3或4的最短路径就是<0,3>或<0,4>吗？ 不能！

D中最短的值就对应了一条最短路径。

第六章 图

第五节 最短路径

- 二、算法
  - 思路？



设S为已求得的最短路径的终点的集合

经过第一步之后：S={1}

下一条最短路径(设其终点为v_i)为以下之一：

1. 中间只经过S中的顶点j而后到达顶点v_i的路径

2. 弧<v_0, v_i>

$$D[i]=\mathrm{Min}\{<v_0,v_i>, D[j]+<v_j,v_i>\}$$

$i\in$ **V-S**

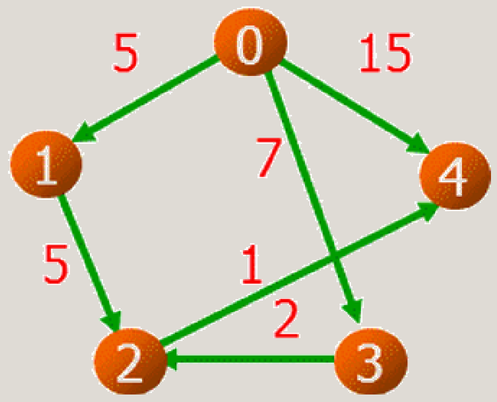
$j\in$ **S**

迪杰斯特拉（Dijkstra）算法

第六章 图

第五节 最短路径

- 二、 Dijkstra算法
  - 简化表达



设S为已求得的最短路径的终点的集合

如果在S中也包含出发点0；并定义D[0]=0。

经过第一步之后：S={0，1}

$$D[i]=\mathrm{Min}\{<v_0,v_i>, D[j]+<v_j,v_i>\}$$

$i \in \mathbf{V\text{-}S}$

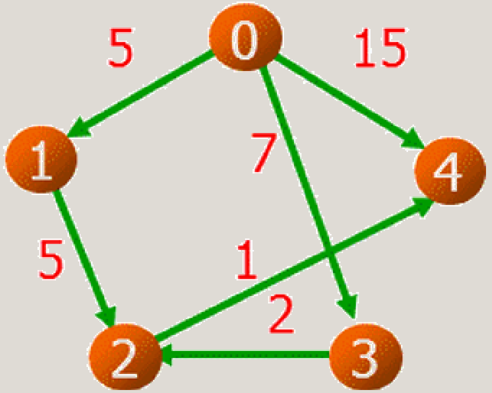
$j \in \mathbf{S}$

$$D[i]=\mathrm{Min}\{D[j]+<v_j,v_i>\}$$

第六章 图

第五节 最短路径

- 二、Dijkstra算法
  - 例子



顶点	当前代价 D[i]			
1	5 {0,1} √	—	—	—
2	∞	10 {0,1,2}	9 {0,3,2} √	—
3	7 {0,3}	7 {0,3} √	—	—
4	15 {0,4}	15 {0,4}	15 {0,4}	10 {0,3,2,4} √
终点j	1	3	2	4
S	{0,1}	{0,1,3}	{0,1,3,2}	{0,1,3,2,4}