

# 第八讲 图（下）

浙江大学 陈 越

# 8.1 最小生成树问题

# 什么是最小生成树 (Minimum Spanning Tree)

最小生成树存在  $\leftrightarrow$  图连通

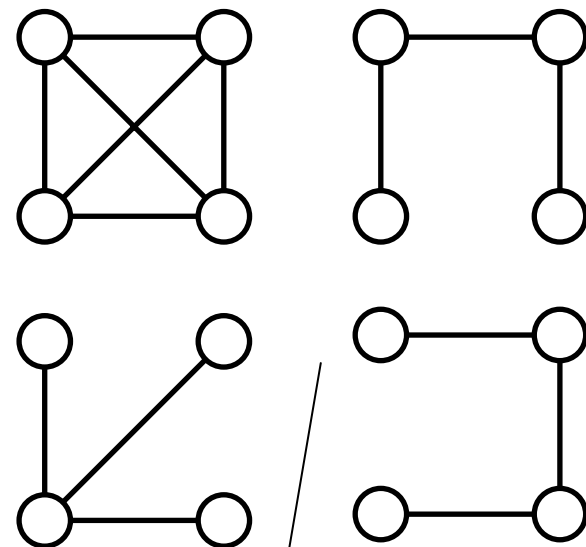
- 是一棵树 树是一种特殊的图, 树是联通的

- 无回路
- $|V|$  个顶点一定有  $|V| - 1$  条边

- 是生成树

- 包含全部顶点
- $|V| - 1$  条边都在图里

- 边的权重和最小

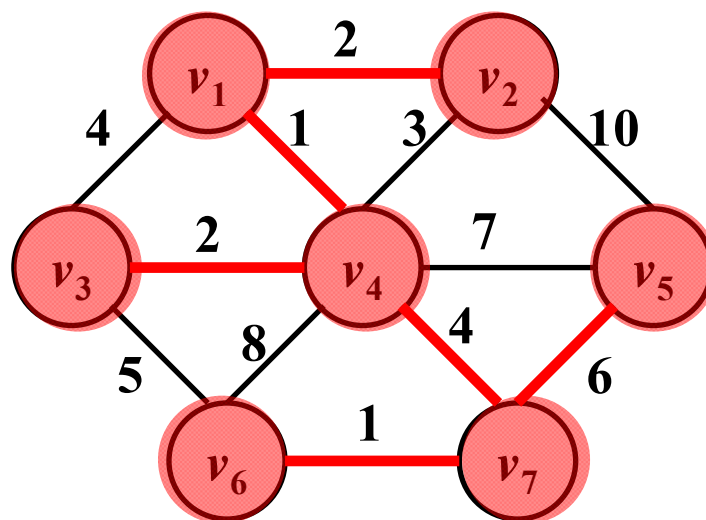


向生成树中任加一条边  
都一定构成回路

# 贪心算法

- 什么是“贪”：每一步都要最好的
- 什么是“好”：权重最小的边
- 需要约束：
  - 只能用图里有的边
  - 只能正好用掉  $|V|-1$  条边
  - 不能有回路

# Prim算法 — 让一棵小树长大



是不是有点像Dijkstra算法.....

# Prim算法 — 让一棵小树长大

```
void Dijkstra( Vertex s )
{ while (1) {
    v = 未收录顶点中dist最小者;
    if ( 这样的v不存在 )
        break;
    collected[V] = true;
    for ( v 的每个邻接点 w )
        if ( collected[W] == false )
            if ( dist[V]+E<v,w> < dist[W] ){
                dist[W] = dist[V] + E<v,w> ;
                path[W] = V;
            }
    }
}
```

dist初始化:  $\text{dist}[V] = E_{(s,v)}$  或 正无穷

对于每一个节点去存  
它父节点的编号:

$\text{parent}[s] = -1$

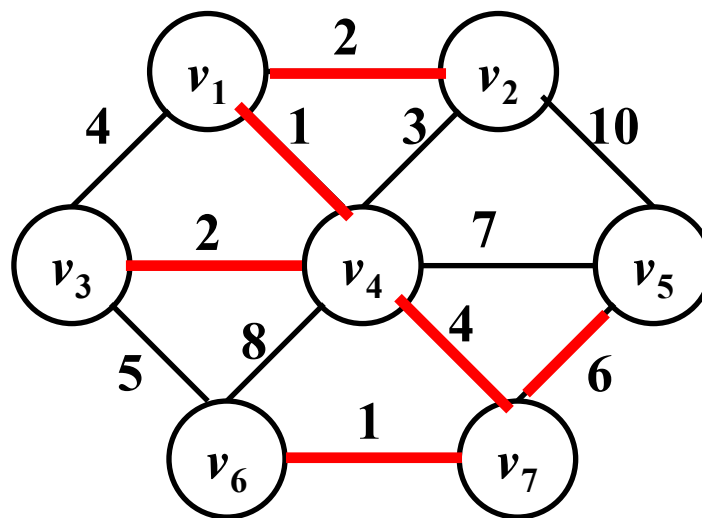
```
void Prim()
{ MST = {s}; /*初始化一颗最小生成树*/
  while (1) {
    v = 未收录顶点中dist最小者; /*dist定义成一个顶点v到这颗生成树的最小距离*/
    if ( 这样的v不存在 )
        break;
    将v收录进MST: dist[V] = 0;
    for ( v 的每个邻接点 w )
        if ( dist[W] != 0 ) /*w没有被收录*/
            if ( E(v,w) < dist[W] ){
                dist[W] = E(v,w) ;
                parent[W] = V;
            }
    }
    if ( MST中收的顶点不到|v|个 )
        Error ( "生成树不存在" );
}
```

$$T = O(|V|^2)$$

稠密图合算



# Kruskal算法 — 将森林合并成树



# Kruskal算法 — 将森林合并成树

```
void Kruskal ( Graph G )
{
    MST = { } ; /*最小生成树里面收集的是边*/
    while ( MST 中不到 |V| -1 条边 && E 中还有边 ) {
        从 E 中取一条权重最小的边 E(v,w) ; /* 最小堆 */
        将 E(v,w) 从 E 中删除 ;
        if ( E(v,w) 不在 MST 中构成回路 ) /* 并查集 */
            将 E(v,w) 加入 MST ;
        else
            彻底无视 E(v,w) ;
    }
    if ( MST 中不到 |V| -1 条边 )
        Error ( “生成树不存在” );
}
```

$$T = O( |E| \log |E| )$$