第八讲 图（下）

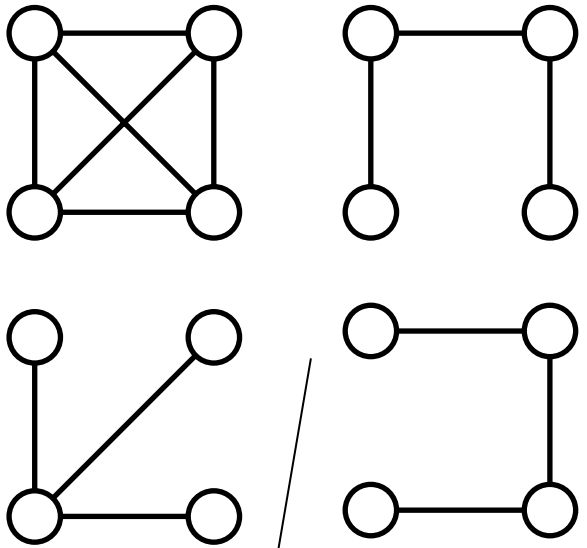
浙江大学 陈 越

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 ↔ 图连通

- 是一棵树
  - 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



这三种都是左上角完全图的生成树

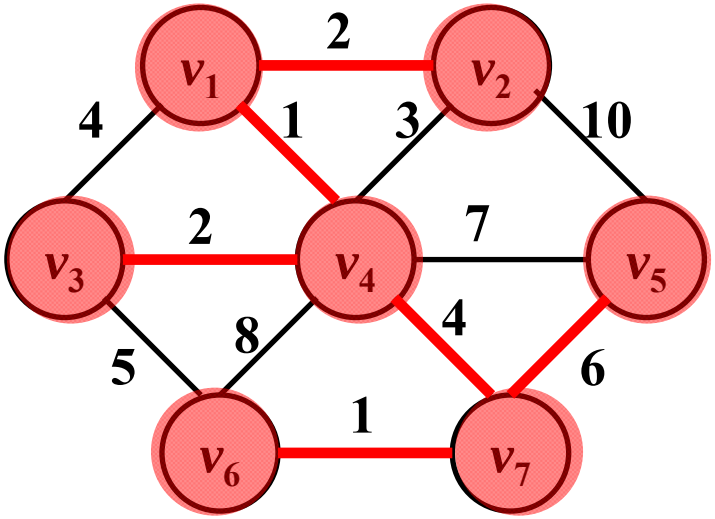
向生成树中任加一条边
都一定构成回路

贪心算法

贪心算法是最小生成树的通用办法

- 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;
         }
  }
}
```

```
void Prim()
{ MST = {s};
  while (1) {
    V = 未收录顶点中dist最小者;
    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 )
         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” );
}
```

说明图是不连通的

dist[V] = $E_{(s,V)}$或 正无穷

parent[s] = -1

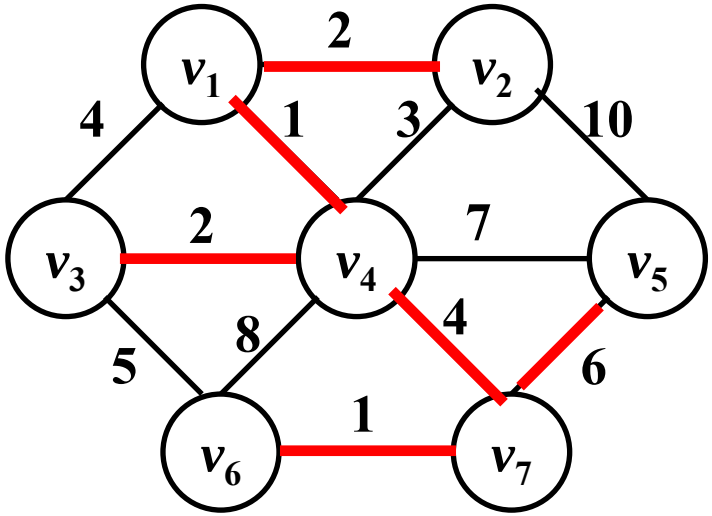
$T = O(|V|^2)$

稠密图合算

Kruskal算法 — 将森林合并成树

稀疏时比prim更高效（边与顶点数差不多）

在满足约束的情况下，每次取权重最小的边，直到满足条件



Kruskal算法 — 将森林合并成树

```
void Kruskal ( Graph G )
{   MST = { } ; 初始为空集
    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” ); 原图不连通
}
```

并查集:如果v,w属于同一棵合并后的树,说明会构成回路

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