

## 8.1.1 Prim算法

---

什么是最小生成树（ ）

1. 是一棵树
  1. 无回路
  2. 边数比顶点数少1
2. 是生成树
  1. 包含全部顶点
  2. 所有的边都在图里
3. 边的权重和最小

向生成树中任加一条边都一定构成回路

最小生成树存在和图连通是等价的

贪心算法

1. 每一步都要最好的
2. 权重最小的边就是最好的
3. 需要约束
  1. 只能用图里有的边
  2. 只能正好用掉 条边
  3. 生成树：不能有回路

Prim算法-让一颗小树长大

```
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")
}
```

稠密图合算

## 8.1.2 Kruskal算法

---

稀疏图合算

将森林合并成树

```

void Kruskal(Graph G)
{
    MST=();
    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_vw;  /* 最小堆 */
        将最小边E_vw删除;
        if(该边E_vw不在MST中构成回路)  /* 并查集 */
        {
            将E_vw加入MST;
        }
        else
        {
            彻底无视该边E_vw;
        }
    }
    if(MST中不到V-1条边)
        ERROR(生成树不存在);
}

```