6.4图的应用(上)

- **最小生成树**
  - 一个连通图的生成树包含图的所有顶点，并且只含尽可能少的边
    - 若砍去它的一条边，则会使生成树变成非连通图
    - 若给它增加一条边，则会形成图中的一条回路
  - 最小生成树：权值之和最小的那棵生成树, 则称为最小生成树
  - 最小生成树性质
    - 最小生成树不是唯一的，即最小生成树的树形不唯一
    - 其对应的边的权值之和总是唯一的，而且是最小的
    - 最小生成树的边数为顶点数减1
  - 最小生成树算法
    - Prim算法
      - 概述
        - 始时从图中任取一顶点加入树T,此时树中只含有一个顶点
        - 之后选择一个与当前T中顶点集合距离最近的顶点,并将该顶点和相应的边加入T,每次操作后T中的顶点数和边数都增 1
        - 以此类推,直至图中所有的顶点都并入T,得到的T就是最小生成树。此时T中必然有n-1条边
      - 时间复杂度 $O(|V|^2)$
      - 适用于求解边稠蜜的图的最小生成树
    - Kruskal 算法
      - 概述
        - 初始时为只有n个顶点而无边的非连通图T={V,{}},每个顶点自成一个连通分量
        - 然后按照边的权值由小到大的顺序,不断选取当前未被选取过且权值最小的边,
        - 若该边依附的顶点落在T中不同的连通分量上,则将此边加入T
        - 否则舍弃此边而选择下一条权值最小的边
        - 以此类推,直至T中所有顶点都在一个连通分量上
      - 时间复杂度 $O(\log|E|)$
      - 适合于边稀疏而顶点较多的图
- **最短路径**
  - Dijkstra算法求单源最短路径问题
    - 辅助数组
      - dist[]:记录从源点v0到其他各顶点当前的最短路径长度,它的初态为:若从v0到vi有弧,则dist[i]为弧上的权值;否则置dist[i]为无穷大
      - path[]:path[i]表示从源点到顶点i之间的最短路径的前驱结点
    - 实现过程
      - 1）初始化:集合S初始为{0},dist[]的初始值dist[i]=arcs[0][i],i=1,2,...,n-1
      - 2)从顶点集合V-S中选出vj,满足 $\mathrm{dist}[j]=\mathrm{Min}\{\mathrm{dist}[i]\mid v_i\in V-S\}$ vj就是当前求得的一条从v0出发的最短路径的终点,令 $S=S\cup\{j\}$
      - 3）修改从V0出发到集合V-S上任一顶点vk可达的最短路径长度:若dist[j]+arcs[j][k]<dist[k],则更新dist[k]=dist[j]+arcs[j][k]
      - 4）重复2）~3）操作共n-1次,直到所有的顶点都包含在S中
    - 时间复杂度 $O(|V|^2)$
    - 不适用于边权值存在负数的情况
  - Floyd算法求各顶点之间最短路径问题
    - 实现过程
      - 初始时,对于任意两个顶点vi和vj,若它们之间存在边,则以此边上的权值作为它们之间的最短路径长度
      - 若它们之间不存在有向边,则以无穷大作为它们之间的最短路径长度
      - 以后逐步尝试在原路径中加入顶点k（k=0,1,2....n-1）作为中间顶点
      - 若增加中间顶点后,得到的路径比原来的路径长度减少了,则以此新路径代替原路径
    - 时间复杂度 $O(|V|^3)$
    - 允许图中有带负权值的边,但不允许有包含带负权值的边组成的回路
    - 适用于带权无向图