1. 稀疏图的定义

若边或弧的个数 e<nlogn，则称作稀疏图，否则称稠密图

1. 图中一个结点的度是指与顶点相关的边的条数，与树结点的度是指出度不同。
2. 图中的路径长度是指：

非带权图：路径上边的条数

带权图：路径上各边的权之和

1. 图的存储结构
2. 顺序存储
3. 邻接矩阵: 对边(或弧)采用二维数组(即矩阵)进行存储
4. 实现方式

顶点数组是一个一维数组；

边，由一个二维数组: 即矩阵(无权图矩阵元素则为0/1，带权图则是权数)。

(无向图)

(带权图)

1. 性质

出度看行，入度看列；

适合稠密图。

1. 邻接表: 对边(或弧)采用链式存储

对顶点依然使用一个一维数组

1. 实现方式

·顶点数组每一个结点包括数据域和一个指针域，指针指向边表的第一个结点

·边或弧用链表的形式进行存储:边结点包括邻接点域和指针域。邻接点域存储邻接点在顶点数组中的下标，指针域指向下一个邻接点。

1. 性质

·无向图中顶点vi的度为第i个单链表中的结点数。

·有向图中(以vi为弧头):

顶点vi的出度——第i个单链表中的结点个数

顶点vi的入度——整个单链表中邻接点域值是i的结点个数(不容易确定，需要遍历整个邻接表)

1. 由于邻接表不易确定入度，因此引入逆邻接表: 弧表记 录以Vi为弧尾的弧。
2. 十字链表: 用于有向图，将邻接表域逆邻接表结合
3. 邻接多重表
4. 图的遍历
5. 深度优先求两点间的简单路径
6. 最小生成树
7. 生成树的概念

连通图中的一个极小连通子图，含有图中全部n个顶点n-1条边

·连通图的生成树并不唯一

1. 最小生成树的概念

·权值最小的生成树称为最小代价生成树

·生成树各边的权值总和称为生成树的权

1. 求最小生成树的两种算法

·prim算法(归并点)

·Kruskal算法(归并边)

1. 拓扑排序(AOV网的应用)

用邻接表的形式存储图结构，增加一个入度数组。

算法逻辑: 初始化入度数组d、初始化栈S→计算各顶点的入度→将

1. 关键路径
2. AOE网的概念

·有向边表示一个工程中的活动 (Activity)

·边上权值表示活动持续时间 (Duration)

·顶点表示事件 (Event)

1. 关键路径

是指活动完成至少需要多少时间，至少需要多少时间往往由最长路径决定，因为需要等所有准备条件完成后才能进行下一步。

即: AOE网中的最长路径就是关键路径

1. 最短路径问题
2. 求单源点的最短路径(一个点到其它所有点的最短路径)

迪杰斯特拉算法(图的存储结构为邻接矩阵):

·算法思想:

贪心算法: 由局部最优推至全局最优。找到一个以直接路径作为最短路径的点(必然存在)，合并它构成”新源点”，则又会出现一个与”新源点”直接路径作为最短路径的点。

1)一个点到另一个点的最短路径只有两种情况: 要么是直接路径，要么是通过其他点到达;

2)与源点直接路径长度最短的邻接点必然是直接路径即为最短路径;

3)接着将第一个找到最短路径的点与源点看作一个整体作为”新源点”，更新最短路径，则又会出现一个与”新源点”路径最短的邻接点，这个邻接点的直接路径一定是它的最短路径，则又可以构成一个新新源点……如此反复。