图

1. 图$G=(V,E)$，V是图中顶点的**非空有限集合**，记为$V(G)$,E是图中边的集合记为$E(G)$
2. 相关名词和概念
    1. 子图，顶点和边的子集
    2. 有向图，无向图，弧，边，弧头顶点，弧尾顶点，有向完全图，无向完全图，度$TD(V)$，入度$ID(v)$，出度$OD(v)$,
    3. $度=出度+入度$
    4. 对有向图来说，有$0\leq e\leq n(n-1)$,对于无向图，有$0\leq e\leq \frac{n(n-1)}{2}$,
    5. 无论是有向图还是无向图，均有$e=\frac{1}{2}\sum_{i=1}^{i=n}TD(v_i)$,
    6. 路径，简单路径(通过任意顶点均不超过一次)
    7. 环，回路，简单环，
    8. 带有权值的图称为带权图或者网
    9. 无向图中，连通图，无向图中的极大连通子图称为连通分量，注意“极大”的理解
    10. 有向图中，强连通图，有向图中的极大强连通子图称为强连通分量
    11. **注意，单一的一个点也是强连通分量**
    12. **邻接点，对于有向图只能说弧头顶点是弧尾顶点的邻接顶点**
3. 图的存储结构
    1. 邻接矩阵
        1. 包括存储顶点信息的顶点表和存储顶点之间相互关系的矩阵，该矩阵称为邻接矩阵
    2. 邻接矩阵表示法有以下的特点
        1. 有向图的邻接矩阵一定是对称的，可以压缩存储节约空间
        2. 图的邻接矩阵表示是唯一的
        3. 对于含有n个顶点的图，邻接矩阵存储时，无论有向图和无向图，无论边的数目是多少，其**存储空间都是$O(n^2)$的，所以邻接矩阵适合存储边数多的图**
        4. 无向图中邻接矩阵第i行（列）非零元素个数是第i个顶点的度
        5. 有向图中，行是出度，列是入度
        6. 邻接矩阵容易判断两个顶点之间是否有边，但是要想得到图中边的总数，则需要遍历，这是邻接矩阵的局限性
    3. 邻接表
        1. 是图的一种链式存储结构，图中所有顶点构成顶点表，每个顶点建立的单链表称为边表
        2. 三种结构，如何正确地使用？，能起到简化代码，梳理逻辑的作用
        3. 邻接表的特点
            1. 对于无向图，顶点V_i的边表中节点的个数为该顶点的度，而有向图中只是该顶点的出度
            2. 与此对应的有逆邻接表
            3. 邻接表表示不唯一，与节点的顺序有关
            4. n个顶点，e条边的无向图，邻接表有n个表头节点和2e个边节点，而对于有向图来说，有n个表头结点和e个边结点，因此**对于边数目较少的稀疏图来说，邻接表相比邻接矩阵要节省空间**
            5. 无向图和有向图中邻接表顶点的个数与度的关系
            6. 十字链表，邻接多重表？？考不考
        4. 图的遍历
            1. 图的遍历与树的遍历的区别
                1. 图中没有唯一的开始结点

    2. 图中只能从一个顶点访问到该顶点所在连通分量上的点，非连通图哈药考虑其他连通分量上顶点的访问
    3. 图的遍历为了防止重复访问的现象，要有一个visit数组
    4. 图的遍历结果不唯一
2. BFS（广度优先遍历）
    1. **如果希望找到一个两个特定顶点之间的一条路径，且边数最短，使用此遍历方法**
    2. 借助队列来实现
3. DFS（深度优先遍历）
    1. 遍历过程的实质是对每个顶点寻找其邻接点的过程，
4. 比较
    1.

BFS	DFS	存储方式
O(n+e)	O(n+e)	邻接表
O(n^2)		邻接矩阵

    2. **两者的区别在于对顶点的访问次序不同**
5. 贪心算法
    1. 贪心算法并不总能够得到最优解，可以在大多数情况下得到最优解的近似解
    2. 活动安排问题贪心的选择活动尽早结束可以得到最优解
    3. 活动安排问题步骤
        1. 排序，加入可行集合
    4. 贪心算法可行的基本要素
        1. 贪心选择性质：指问题的整体最优解可以通过一系列的局部最优解得到，即贪心选择来达到
        2. 最优子结构性质：指问题的最优解包括子问题的最优解？？？和递归的比较
        3. 如何理解?
    5. 贪心算法的应用
        1. 求解带权最短路径问题
            1. Dijistra算法
                1. 最短单源路径寻路
                2. 复杂度为O(n^2)
        2. 最小生成树问题
            1. 定义：有n各顶点的连通图的生成树是一个极小连通子图，包含所有顶点，且只含有n-1条边
            2. 如果在最小生成树上添加一条边，一定会构成环
            3. 特点
                1. 任意两个顶点之间有且只有一条路径
                2. 加上一条边一定出现环，删去一条边。一定变成非连通图
                3. 一个图可以有多个生成树，只要一个最小生成树
                4. 一个有n各顶点的完全图中有n（n-2)中不同的生成树
                5. 生成树可以通过图的遍历得到
                6. 深度优先生成树，广度优先生成树
            4. 切分是图中顶点的一个子集，切分集是图中边的一个子集
            5. 切分性质，回路性质

6. Prim算法
    1. 复杂为O(n^2)，它与带权无向图的边数无关，适合求边稠密的无向图的最小生成树
7. 克鲁斯卡尔
    1. 复杂度为O(eloge)，适合求边稀疏的图的最小生成树
8. AOV网
    1. 顶点称为活动，
    2. 拓扑排序，拓扑序列（不唯一）
    3. AOV网中不能出现回路
    4. 拓扑序列的构建方法
        1. 找无前驱的节点
        2. 删除该节点和相关的边
    5. 拓扑排序过程中需要容易找到顶点的入度，需要容易寻找任意顶点的所有直接后序
    6. 复杂度O(n+e)
9. AOE网，
    1. 关键路径（不唯一）
    2. 算法?
10.