

分类：

有向图、无向图

conception：

1、顶点、弧：

弧头——终端点

弧尾——初始点

2、完全图：

n个顶点有向图：n(n-1)条边

n个顶点无向图：n(n-1)/2条边

3、连通图：

无向图——每两个点可以相互到达

有向图，分为：

强连通图——双向均可到达

弱连通图——单向到达即可

---》强连通分量

连通分量——**极大**连通子图（非连通图可能有多个连通分量）

生成树——**极小**连通子图

4、度：

有向图——入度、出度

5、稀疏图、稠密图（针对弧来说）

6、重数

连接两个相同顶点的边的条数,叫做边的重数

连通分量&连通子图

- 连通图的极大连通分量即它自己，只有一个
- 生成树是极小连通子图，可能不止一个；
- 只有连通图有极小连通子图（生成树）
- 而连通分量是极大连通子图
- 非连通图有多个极大连通子图。（非连通图的极大连通子图叫做连通分量，每个分量都是一个连通图）
- 非强连通图有多个极大强连通子图。（非强连通图的极大强连通子图叫做强连通分量）
- 任何连通图的连通分量只有一个，即是其自身，非连通的无向图有多个连通分量。

遍历方式：

dfs——树的先根遍历

bfs——树的层序遍历

连通性问题：

无向图的连通分量

DAG有向无环图：

有向图的强连通分量

关节点&重连通分量

关节点——删除该顶点及其边后，其所在的的连通分量变为两个以上

重连通分量——不存在关节点

拓补排序：

偏序&全序

两种DAG图的应用：

AOV网——点表示活动、过程、工序..., 边表示工序间的次序关系

AOE网——边表示活动、过程、工序

存储方式：

图没有顺序映像的存储结构，但可以借助数组的数据类型来表示1元素之间的关系

数组表示方式：

对于图：用Map[i][j]=0/1表示 顶点i 到 顶点j 是否存在边

对于网：用Map[i][j]=INF表示 顶点i 到 顶点j 不存在边

用Map[i][j]=w 表示 顶点i 到 顶点j 存在权值为w的边

网——有权值的图

邻接表：（是种链式存储结构）

对于n个顶点、e条边的**有向图**——n个顶点结点，e个边结点

对于n个顶点、e条边的**无向图**——n个顶点节点，2e个边节点

有两个node类型——表结点（3个域）、头节点（2个域）

表结点：邻接点域、链域（下一条边）、数据域

边结点：data（链表相关信息）、第一条边

十字链表：

仅针对**有向图**——每个弧节点有5个域，其弧头和弧尾顶点结点 及 弧头和弧尾的下一条弧结点

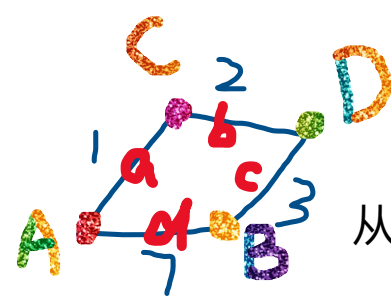
邻接多重表：

仅针对**无向图**——每条边**只要一个边结点**来存储，每个弧结点

有6个域（类似5个域，多一个用来标记是否被搜索过）

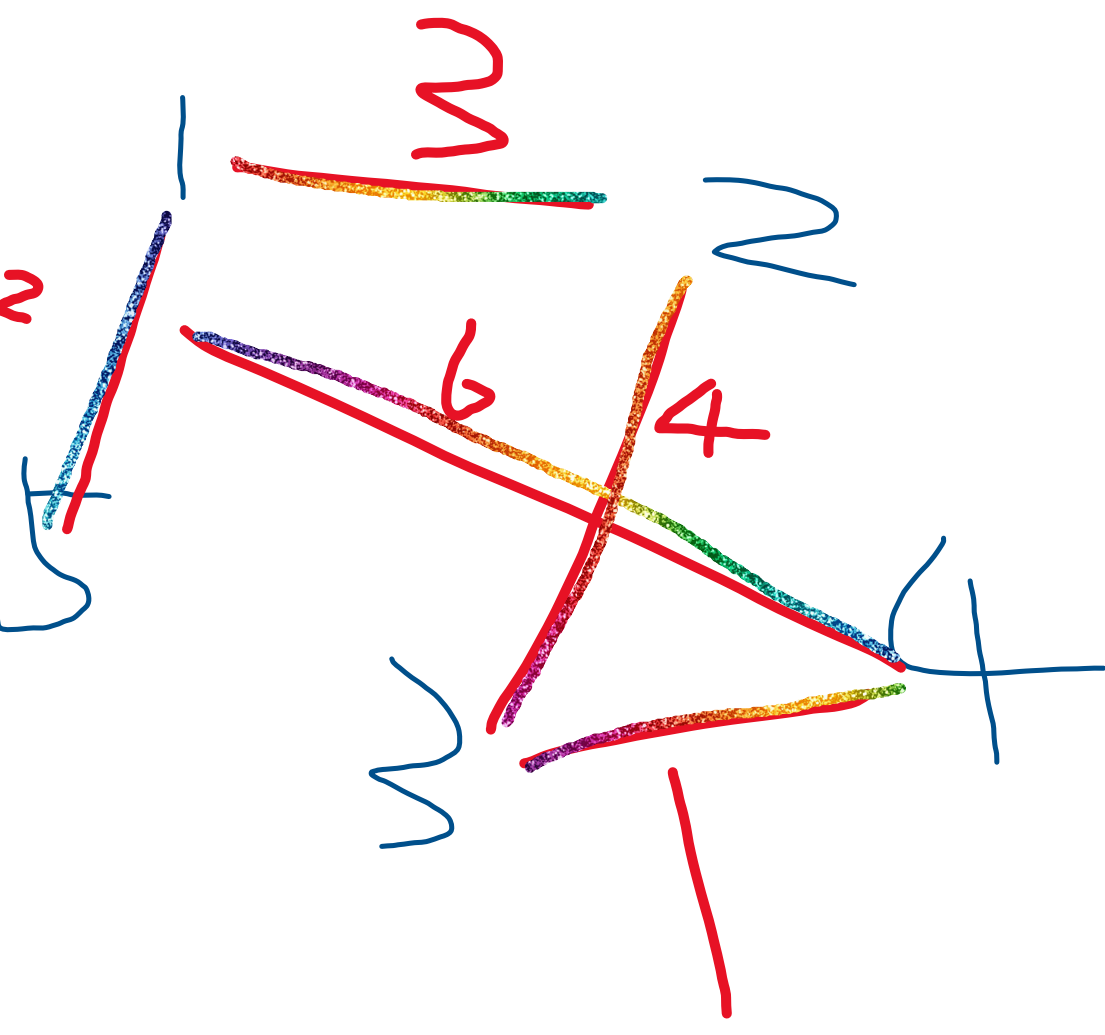
最短路径：

单源最短路径：

- Dijkstra：
 - 原理：从A->B选取abc路径：必有a<d，可以确定边a为A与C间的最短路
 - 限制条件：图G中不存在负权值的边
 - 两件事：
 - 【1】不断运行广度优先算法找可见点，计算可见点到源点的距离长度（可见点就是从源点开始按广度优先算法遍历顶点的过程中，搜索到的点）
 - 【2】从当前已知的路径中选择长度最短的将其顶点加入S作为确定找到的最短路径的顶点

所有顶点间的最短路径：

- Floyd：
 - 限制条件：图G中可以存在负权值的边，但是不可以有存在负值的环



5
1 2 3
2 3 4
3 4 1
4 1 6
1 5 2
2 5