1.图的基本概念

图论，在数据结构中占据着相当重要的地位

因为在实际开发中，图论中涉及到的各种问题，都有很直观的应用。

例如，在图论中有一个问题就是多源最短路径问题，而这个问题在实际开发中最直观的表现就是各种地图软件的底层原理

再比如，图论中的关键路径问题，在工程控制和软件工程方面，能够用来运算项目中的那些模块在开发工期上可以延迟和提前，而哪些是不可以的

所以，图论问题在真实开发中的应用是十分广泛的，这也奠定了图论在数据结构方面的重要地位

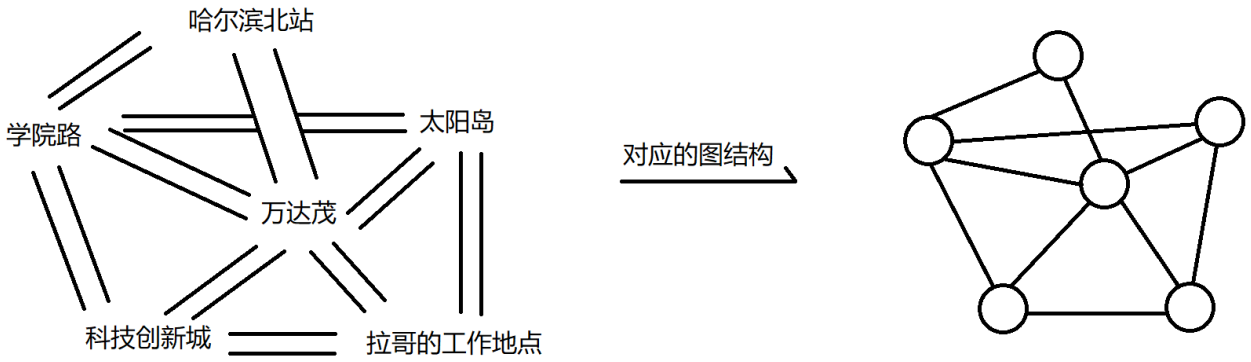
在开始学习图论之前，先让我们来了解一下图论中会使用到的基本概念

1.图

图是什么？用书本上的方式来进行解释，就是节点和边的集合（抱歉我没理解上去……）

图就是一些节点，通过边连接起来所构成的一种结构。节点就好比地图上的建筑物，边就是地图中的条条大道

这些道路将建筑物连接起来，就构成了地图；同理，使用边将节点连接起来，就构成了图论中的图



2.节点

在上面图的概念中，我们提到了节点的概念，实际上我们可以将图论中的节点看做是地图中的建筑物

节点也是构成图的基本概念之一

3.边

在图中，将所有节点两两连接在一起的，就是边，我们也可以认为边就是从一个地点到达另一个地点之间的道路

但是和现实生活中不同的是，现实生活中的道路，往往是可以拐弯的，但是在数据结构中，为了更加直观的表现两个节点之间的联系，往往使用直线来表示边

图论中的边，一般不会使用曲线表示（一些特殊情况除外），并且一条边的两端，一定都是节点

如果两个节点之间存在一条直接的边，那么我们认为这两个节点之间是可达的，也就是存在通路的

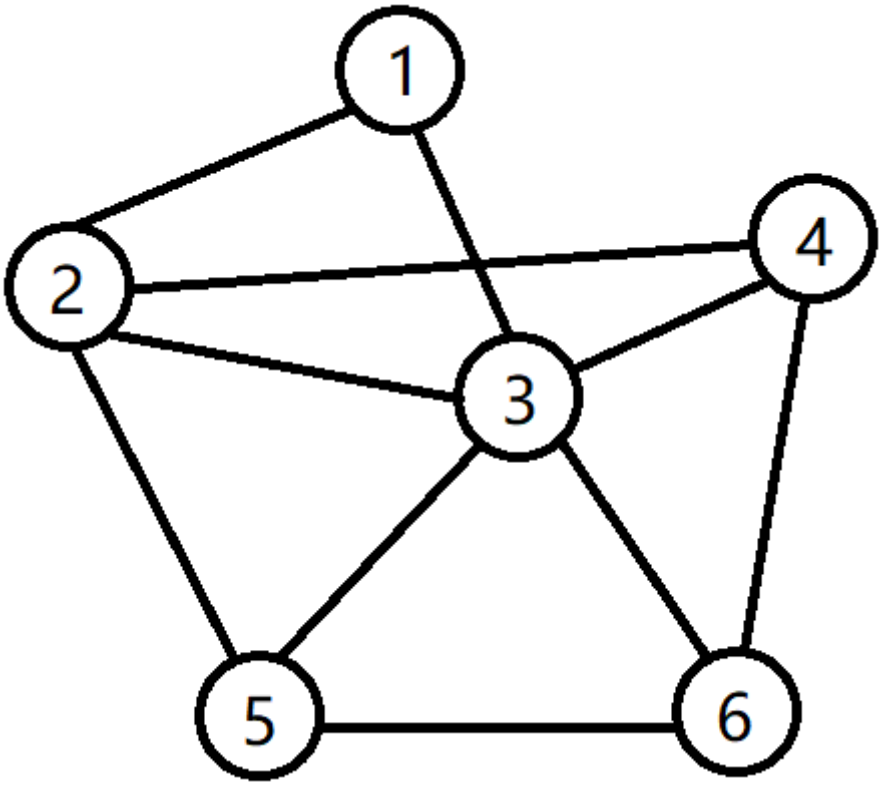
边与边之间是可以交叉的，就好比现实生活中的立交桥一样，可以上下层叠

4.无向图

在一张图中，如果存在联通所有节点的边是没有方向指向的，那么这种图就称之为无向图

我们也可以认为，在无向图中，来往同一条边的两个端点之间的路只有一条，来去都走这条路，而且路的长短是不变的

一个无向图：



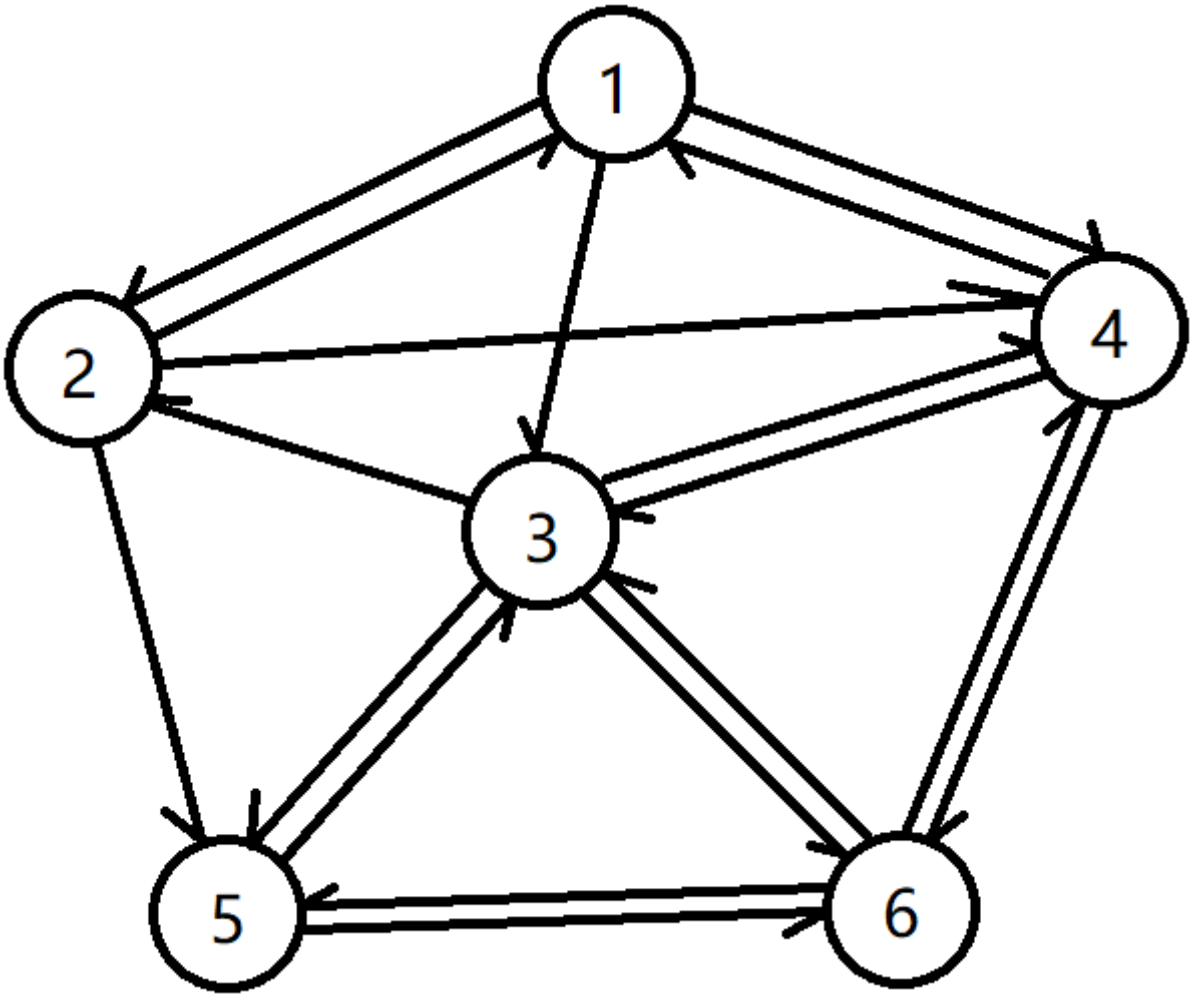
5.有向图

有向图的定义就相对比较麻烦了，因为在有向图中，从A点到达B点，和从B点到达A点之间的路径，因为方向的不同而不是同一条路

我们可以做如下描述：从火车站到宾馆的路上是直的，没有任何障碍；但是在从宾馆到火车站之间的对向车道上有个钉子户占道，所以我们就要绕一些远了

那么在有向图中，有的节点可以双向通行，有的节点之间只能够单向通行（例如单行道），而双向通行的道路之间，来回的距离还可能是不一样的

一个有向图：



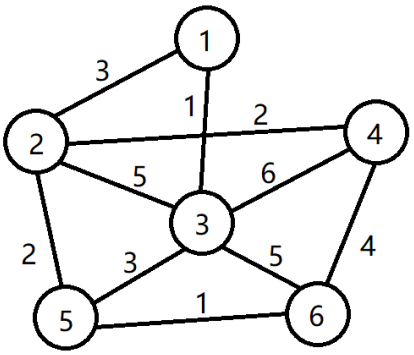
6.边的权值

如果在图（包括有向图和无向图）中，如果一个边上，通过数字来表示两个节点之间的相对距离，那么这个图就变成了带权图

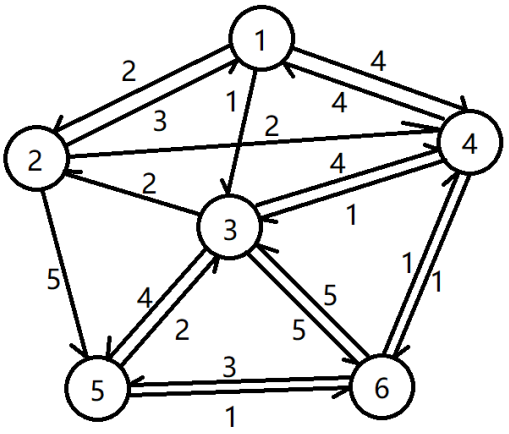
边上用来表示相对距离的数组，就是边的权值，简称边权

通过边权和上面说到的有向图和无向图的概念，我们也就可以得到有向带权图和无向带权图的概念

在无向带权图中，每一对连通的节点之间来回的举例，也就是边权是固定的；而在有向带权图中，因为两个节点之间来回的边不是同一条边，所以也可能具有不同的边权



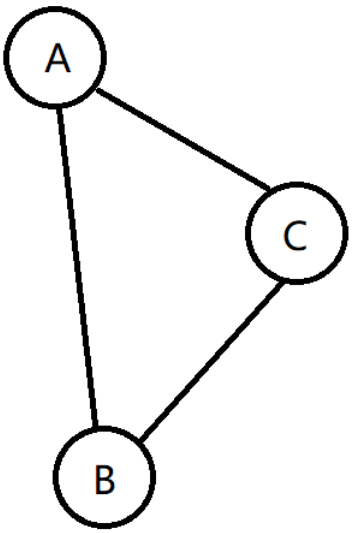
无向带权图



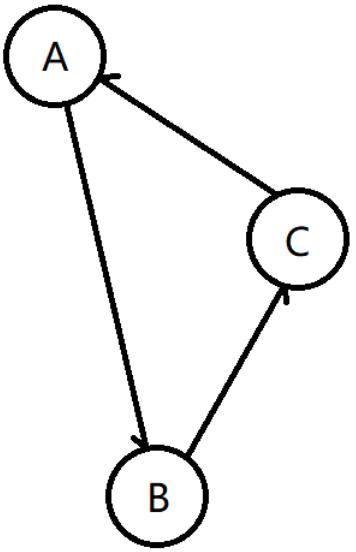
有向带权图

7.回路和环路

回路的概念在有向图和无向图中都是存在的：如果一个节点从自身出发，可以通过n个节点和n-1条边最终回到自己身上，那么这些节点和边就构成了一个回路



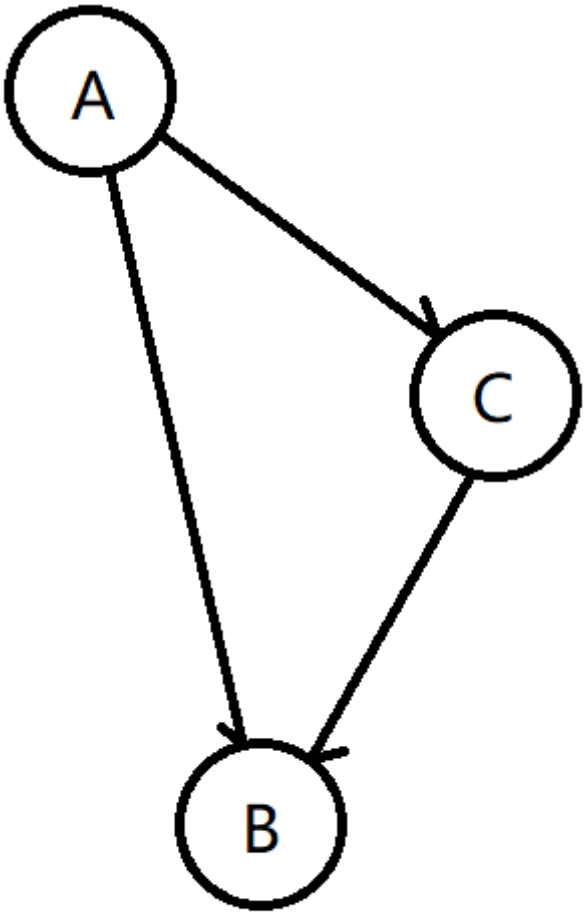
无向图的回路



有向图的回路

一种特殊的回路

而环路的概念，一般专指有向图：如果n个节点之间使用n-1条边相连，但是从起点出发，最终不能回到终点，那么就构成了一个环路



有向图的环路

总结起来就是：

回路：兜兜转转，回到起点

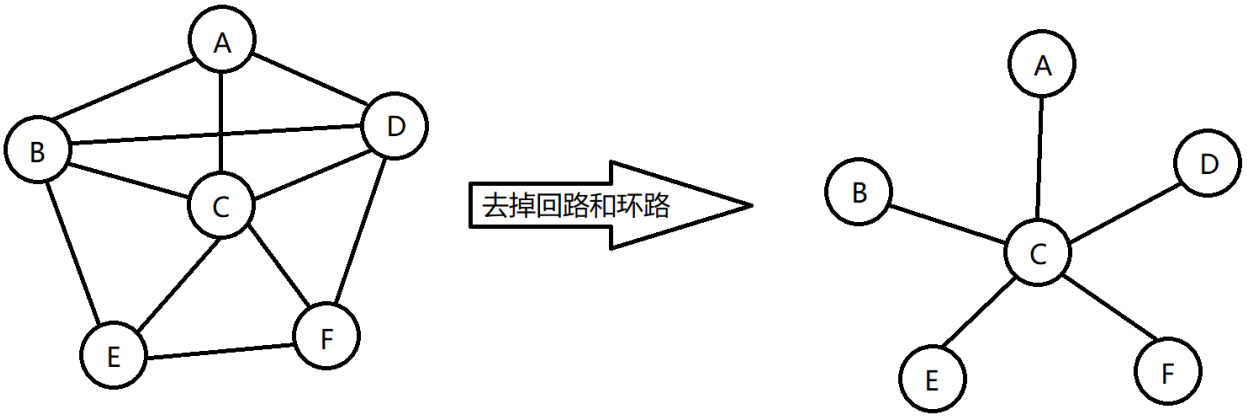
环路：山重水复，殊途同归

8.用图的概念定义树

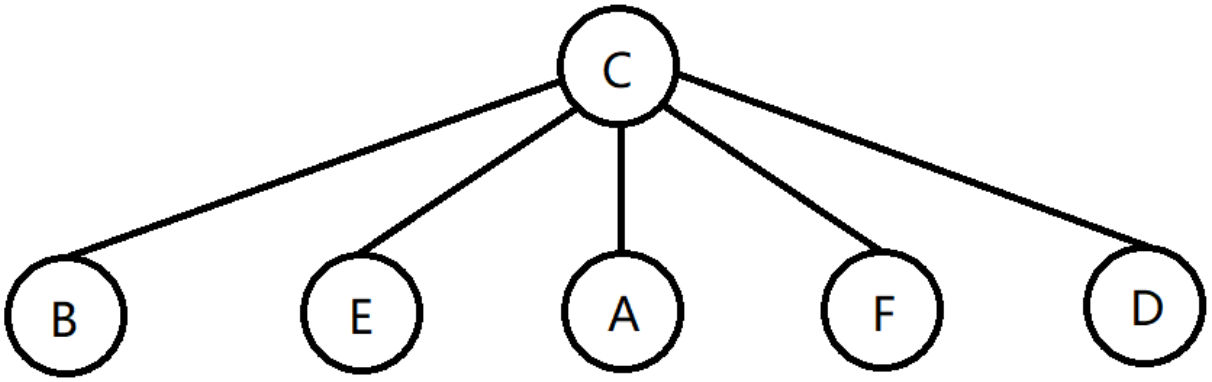
在前面二叉树的章节中，我们曾经介绍过树的官方定义：没有回路和环路的图（为啥官方定义永远这么迷……）

现在我们已经了解了什么是图、回路和环路的概念，那么我们试着使用官方定义来解释一下什么是树

如下图所示，我们将一个图中的回路和环路全部去掉：



然后再调整一下节点之间的角度：



一个树结构就出现了