路程规划问题

暑假到了，同学们都准备好暑假的出行计划了吗？怎样做一个最好的出行计划的呢？今天我们就来学习这类问题的解决方法。除了路程规划，现实生活中还有很多这类问题（比如导航设计、AlphaGo等棋类游戏AI的最优步数计算、迷宫设计和解答、电路板设计等等），学习它们的解决方法可是大有作用的呢。

假如住在广州，准备坐高铁或火车到重庆去。下面是一个中国内地大城市的铁路图 – 请问经过的站点数最少的路程是哪一条呢？



为了找到经过的站点数最少的路线，我们可以这样思考：

广州坐1个站可以到达的城市有哪些？很明显，有深圳和长沙。--重庆不在里面，所以一个站到不了重庆。

广州坐2个站可以到达的城市有哪些？深圳出发坐一个站不包括广州只有福州；长沙出发坐一个站不包括广州有贵阳、杭州、武汉。所以广州坐2个站可以到达的站点包括福州、贵阳、杭州、武汉，还是到不了重庆。

广州坐3个站可以到达的城市有哪些？根据上面的规律，我们找到武汉坐一个站可以到合肥、郑州、重庆。耶，重庆出现了。所以我们知道广州到重庆站点数最少的路线是广州-长沙-武汉-重庆，一共要坐三个站。

类似这种找经过最少站点的问题叫做**最短路径问题。**这是生活中经常会遇到的问题，除了最常用到的出行规划，你还想到什么类似的问题吗？下面这些都是最短路径问题：

国际象棋、围棋等计算把对方将死的最少步数。还记得鼎鼎大名的AlphaGo吗？

人脉关系度数计算 -- 有一个六度人脉关系理论：地球上所有人都可以通过六层以内的熟人链和任何其他人联系起来。就是说，最多通过六个人你就能够认识任何一个陌生人。微博等社交软件就是通过最短人脉关系来计算和推荐内容给用户。

解决如何走出迷宫的问题。

电路的设计问题。

等等

我们上面用到的解决方法叫做广度优先算法或宽度优先算法。它的过程可以看成是一个构造一颗倒立的大树的过程：

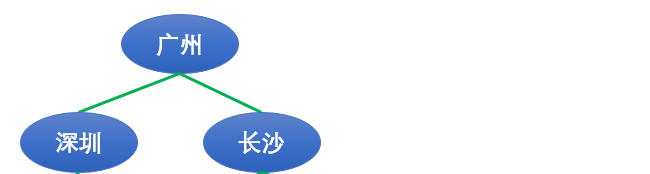
1. 先画根节点，即开始的点
2. 接着依次访问所有未被访问的邻节点，把它们画成树叶并和根节点连起来
3. 检查树叶里是否有要找的终点，如果没有的话，重复此过程，直到所有节点都被访问完。

我们用找广州到重庆的最小路径方法来示范:

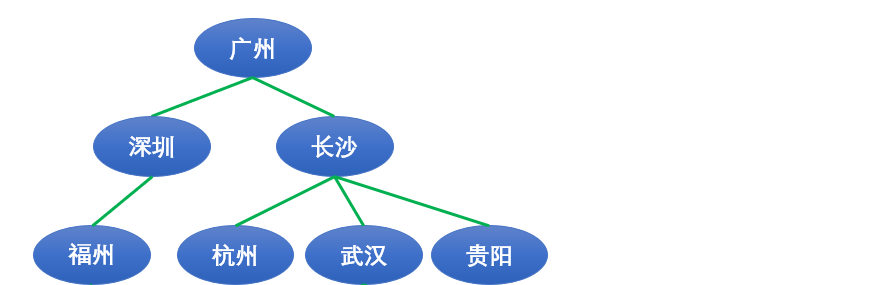
1. 画根节点：广州



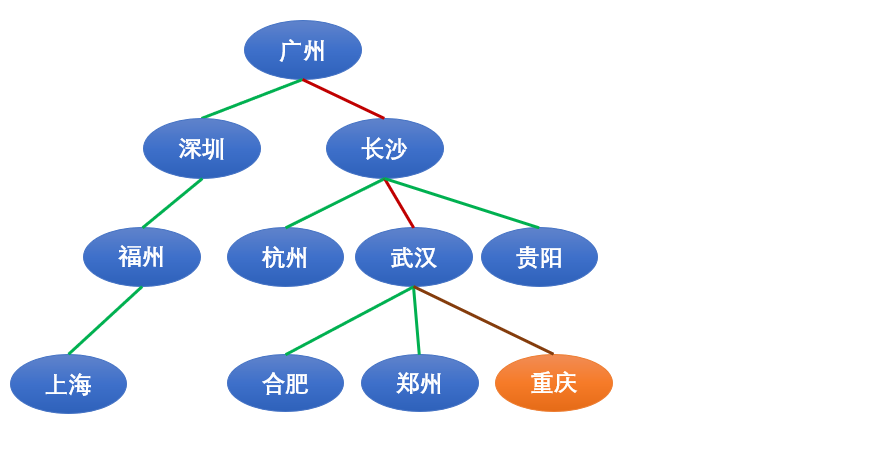
1. 画第一层树叶：昆明未被访问的邻节点 – 深圳和长沙



1. 重庆还没有出现，继续画下一层树叶：深圳的未被访问的邻节点是福州；长沙的未被访问邻节点有杭州、武汉、贵阳



1. 重庆还没有出现，继续画下一层树叶：福州的未被访问邻节点是上海；杭州的邻节点都访问过了；武汉的未被访问邻节点有合肥、郑州、重庆。重庆出现了！树的构造也到此为止，最短路径也找到了：

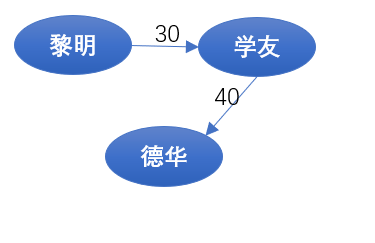


通常将这种由若干个顶点以及连接某些顶点的边所组成的图形称为图。这种图形通常用来描述事物之间的某种特定关系，用点代表事物，用连接两点的线表示相应两个事物间具有这种关系。

这里再介绍几个常见的和图有关的名词：

**有向图**

如果顶点之间的边是有方向，表示一种先后顺序或依赖关系等，则称为有向图，比如黎明欠学友30块钱，学友欠德华40块钱，可以用下面的有向图表示，注意边的箭头：



**无向图**

相反，顶点之间的边是没有方向关系的，则称为无向图。比如我们的铁路图的例子里，相邻城市之间的边（铁路）是没有方向的，广州到深圳和深圳到广州的意义（站点数）是一样的。



**权重**

如果我们给边加上一些特别的数字描述，比如下面的图里我们给每个相邻站点之间的边加上一个数字代表列车在它们之间通行的时间（小时），则这些数字称为权重。不带权重的图称为非加权图。带权重的图称为加权图。我们一开始的线路图是非加权图。下面这个是加权图：



**图算法**

解决图问题相关的算法叫做图算法。

我们上面介绍的广度优先算法就是一种图算法，它可以用来解决非加权图最短路径问题。那么对于加权图怎么办呢？比如我们现在要找从深圳到上海的**时间最短**的路径，如果用广度优先算法，得出来得结果是深圳-福州-上海。但是这条路需要2+5+7=14小时，而如果我们选深圳-广州-长沙-杭州-上海的路径，只需要2+2+5+2=11小时。虽然经过的站点数多，但时间反而省了3个小时。

处理加权图最著名的一种算法叫做Dijkstra算法，一般翻译为迪杰斯特拉算法。它的过程总结来说主要有四个步骤：

1. 找到起始节点最近的一个邻节点

以图为研究对象有专门的一个学科分支叫做图论，并且图也是拓扑学的研究范畴。