# 最短路径-迪杰斯特拉算法

1.高德地图相信大家都知道，里面一个最基础的功能就是最优路线：比如路径最短，时间最短等，你有想过它是用了什么数据结构什么算法？如何来实现的呢？

DFS：

图：

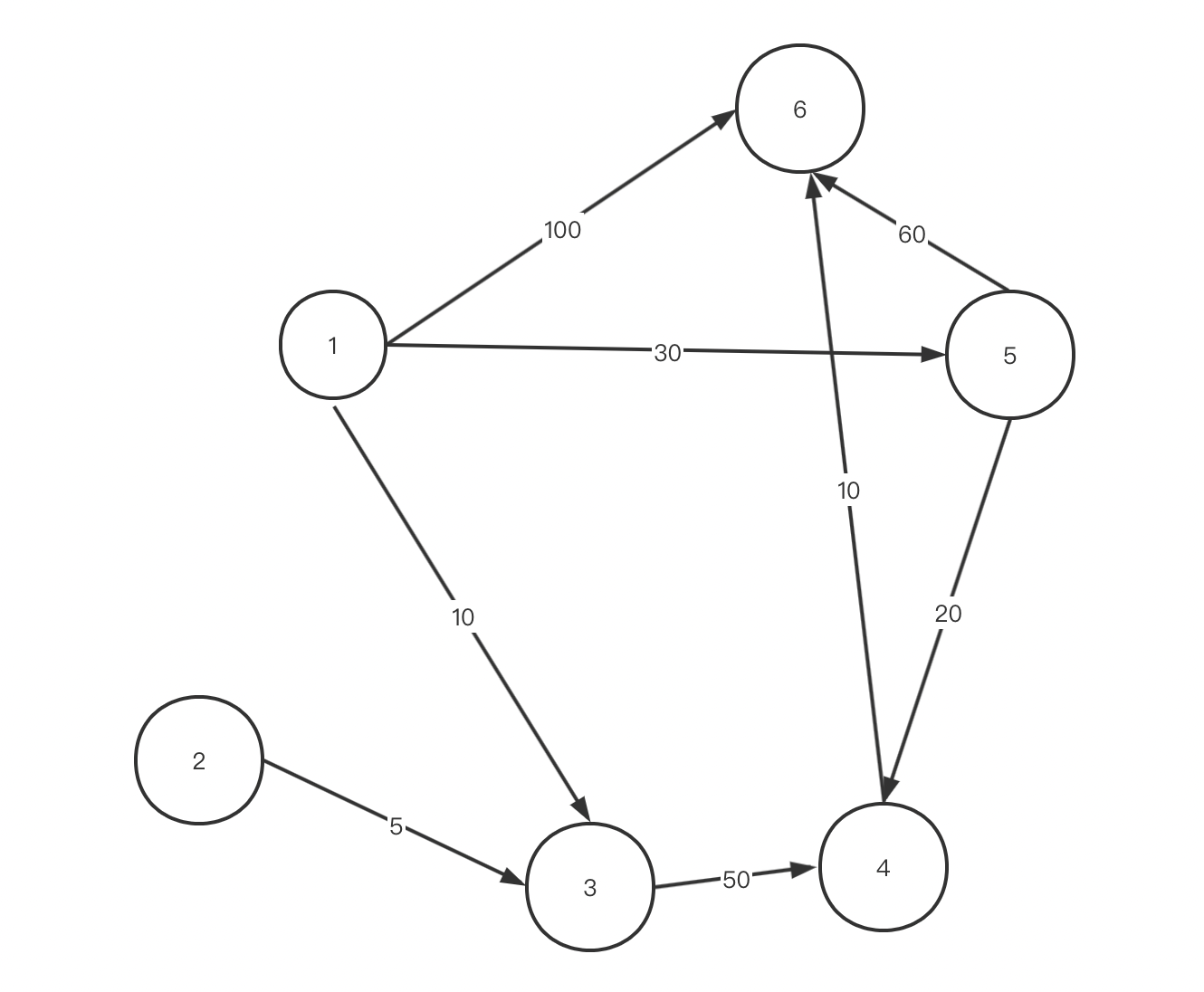
线性表：数组+链表

非线性表:树（）+图

最短路径分析：

说到的地图计算最短路径，首先我们要解决这个问题就要找准一个数据结构，很显然用图结构来表示最好了。

具体：我们可以把每个路口看成一个点，路口之间的路看作一条边。路的长度就是边的权重，即可得到以下这个图形：



这里我们就将地图转换成了我们熟悉的数据结构图，那么假设我要从1点作为起点，则就变成求1点到其他点的最短路径。

动态规划：

贪心算法：局部最优推出全局最优。求1->6的最短路径,排序。

最短路径分析：

1.根据图存储结构大家想想这个我们应该选用什么结构？大图一定要用邻接表(小图可以用邻接矩阵，写法相对简单)。我今天就用邻接矩阵

2.算法选择：经典算法 迪杰斯特拉（Dijkstra）算法，即单源最短路径算法，它是所有最短路径算法的基础，我们的地图软件最终使用的算法也是以他为基础进行的优化。所以弄懂它变得尤为重要。

3.算法核心思路分析：

问题：求1到其他任意点的最短路径。

最短路径分析：

核心思想分析：贪心：排序，贪心策略。

1.我们开一个dis数组，用来表示起始点到每个顶点的距离，最开始时我们赋值为无穷大。

2.加变量loc，初始赋值为起始点。（从哪个点出发就是哪个 举例1）

贪心的策略：在dis数组里面找离初始点最近的那个点

3.通过loc更新dis数组，因为加入一个点后我们就可以更新路径

4.在dis数组里面找离初始点最近的那个点，排除已经选择过的点，将之赋值给loc。

5.重复执行 3 4操作，直到所有点加完。

核心操作是第三步：我们应该如何来更新dis数组呢？

类似于背包问题，每增加一个点进入，取最短的路径长度值放到dis数组里

过程如下：

