**最短路径算法**

解决最短路径问题：（如下三种算法）

（1）迪杰斯特拉算法（Dijkstra算法）

（2）弗洛伊德算法（Floyd算法）

（3）SPFA算法

第一种算法：迪杰斯特拉算法（Dijkstra算法）

广度优先搜索解决赋权有向图或者无向图的单源最短路径问题.是一种贪心的策略

算法的思路

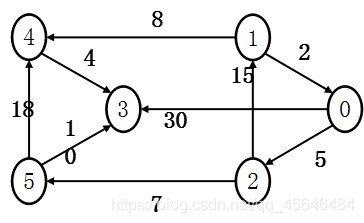
1.将图上的初始点看作一个集合S，其它点看作另一个集合

2.根据初始点，求出其它点到初始点的距离d[i] （若相邻，则d[i]为边权值；若不相邻，则d[i]为无限大）

3.选取最小的d[i]（记为d[x]），并将此d[i]边对应的点（记为x）加入集合S（实际上，加入集合的这个点的d[x]值就是它到初始点的最短距离）

4.再根据x，更新跟 x 相邻点 y 的d[y]值：d[y] = min{ d[y], d[x] + 边权值w[x][y] }，因为可能把距离调小，所以这个更新操作叫做松弛操作。

5.重复3，4两步，直到目标点也加入了集合，此时目标点所对应的d[i]即为最短路径长度。



以这张图为例

首先求出长度最短的一条最短路径：0 -> 2，

长度为 5（2 号点）.顶点 2 的最短路求出来之后，

0到其他顶点的最短路径长度有可能要改变，比如 0 -> 1 从原来的 ∞ 变成了 20， 0 -> 5 从原来的 ∞ 变成了 12.

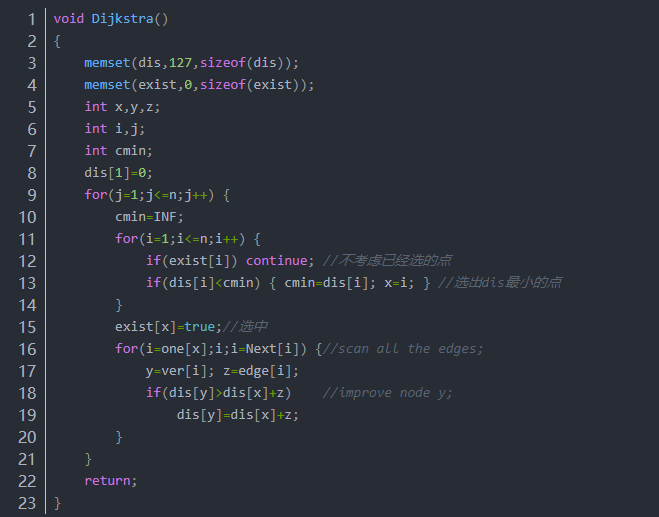
这样次短的最短路就是 12（5 号点）.

5 的最短路求出后，0 到其他点的最短路还会发生改变。比如 0 -> 3 从原来的 30 变为了 22，0 -> 4 也从 ∞ 变为了 28。这时第三短的最短路就确定了，是 20（1 号点）.

之后再依次确定第四短的最短路是 22（3 号点）.

第五短的最短路是 28 （4 号点）

普通模版：



第二种算法：弗洛伊德算法（Floyd算法）

原理：

Floyd算法（弗洛伊德算法）是一种在有向图中求最短路径的算法。它是一种求解有向图中点与点之间最短路径的算法。

用在拥有负权值的有向图中求解最短路径（不过不能包含负权回路）

在Floyd算法中，需要两个矩阵，一个是邻接矩阵A，最开始，A[i][j]的值为节点i到节点j的边的权值，如果i到j没有边，A[i][j]=\infty，特别地，A[i][i]=0。另一个矩阵为P，P[i][j]的值为从i到j的路径上节点i的下一个节点，通过该矩阵可以找出任意节点i到节点j的最短路径。最开始，若i与相邻,则P[i][j]=j，否则P[i][j]=-1。

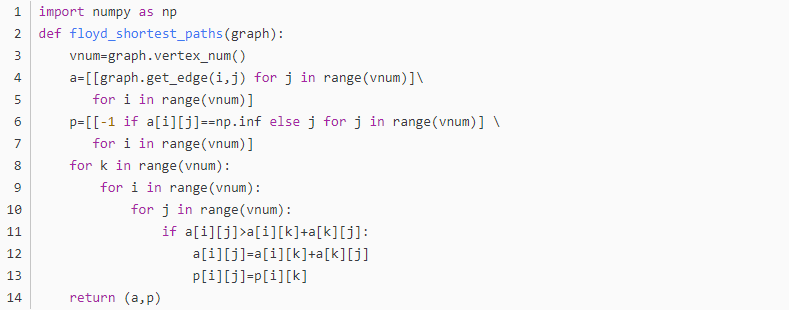
之后将对A和P进行n次迭代更新过程，n为节点个数，迭代更新原则是:

A_{k+1}[i][j]=min{}\left \{ A_{k}[i][j],A_{k}[i][k]+A_{k}[k][j] \right \}(0\leqslant k\leqslant n-1)

如果A_{k+1}[i][j]=A_{k}[i][k]+A_{k}[k][j]，那么P_{_{k+1}}[i][j]=P_{_{k}}[i][k]。

最终得到的A[i][j]就是i到j的最短路径长度,P[i][j]是i到j的最短路径上,i节点的下一个节点。

Floyd算法的python代码：



第三种算法：SPFA算法

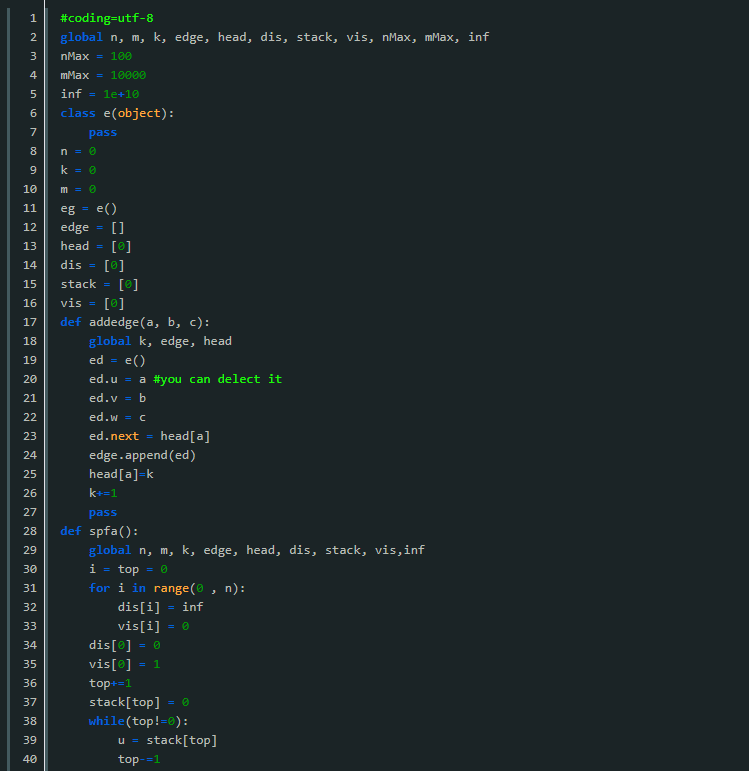
SPFA算法是求解单源最短路径问题的一种算法，由理查德·贝尔曼（Richard Bellman） 和 莱斯特·福特 创立的。有时候这种算法也被称为 Moore-Bellman-Ford 算法，因为 Edward F. Moore 也为这个算法的发展做出了贡献。它的原理是对图进行V-1次松弛操作，得到所有可能的最短路径。

**其优于迪科斯彻算法的方面是边的权值可以为负数、实现简单，缺点是时间复杂度过高，高达 O(VE)。但算法可以进行若干种优化，提高了效率。**

思路：

我们用数组dis记录每个结点的最短路径估计值，用邻接表或邻接矩阵来存储图G。我们采取的方法是动态逼近法：设立一个先进先出的队列用来保存待优化的结点，优化时每次取出队首结点u，并且用u点当前的最短路径估计值对离开u点所指向的结点v进行松弛操作，如果v点的最短路径估计值有所调整，且v点不在当前的队列中，就将v点放入队尾。这样不断从队列中取出结点来进行松弛操作，直至队列空为止。

实现代码：





GPS车载导航最短路径算法的实现：

对最短路问题的研究早在上个世纪60年代以前就卓有成效了,其中对赋权图的有效算法是由荷兰著名计算机专家E.W.Dijkstra在1959年首次提出的,该算法能够解决两指定点间的最短路,也可以求解图G中一特定点到其它各顶点的最短路。后来海斯在Dijkstra算法的基础之上提出了海斯算法。但这两种算法都不能解决含有负权的图的最短路问题。因此由Ford提出了Ford算法,它能有效地解决含有负权的最短路问题。

GPS车载导航最短路径算法实现步骤 ：

（1）跟据location()方法找到主要建筑以及十字路口的坐标并将其坐标记录。

（2）根据void Graph::mouseDoubleClickEvent用鼠标的双击事件找到地图中目的地的坐标。在这里出发地的坐标通过模拟输入的方式获得。

（3）根据出发地和目的地之间的主要预设点信息，一次对点与点直接的距离进行筛选，找到出发地和预设点，预设点和预设点以及预设点和目的地之间最短的线路，并调用Qline方法连线，从而找到最短的路径。

（4）根据不同连线的起点和终点坐标，算出x与y之间的函数关系，使得起始坐标不断沿着找到的最短路径移动，实现导航。

最终结果是，通过各个节点之间的最短路径找到了出发地和目的地之间的最短线路。