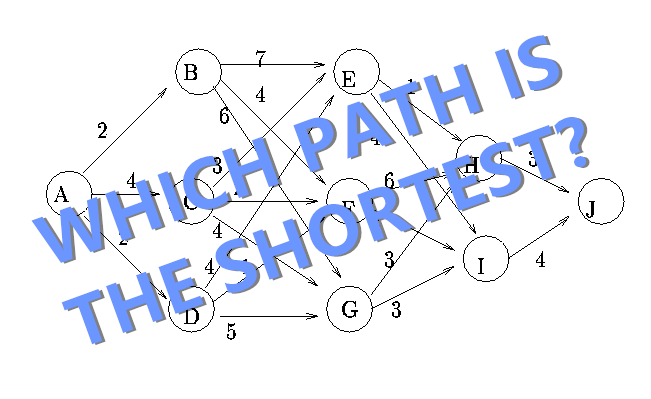
**最短路算法之Dijkstra**

【Background Info】

给一幅图，求点之间的最短路径，是OI中经常遇到的问题。



求最短路径可以用：DFS, Dijkstra, Floyd, Bellman-Ford, SPFA （即Bellman-Ford队列优化）等方法。

今天主要介绍一下Dijkstra。

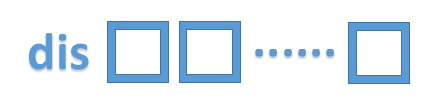
Dijkstra是一个由荷兰计算机科学家Edsger Wybe Dijkstra发明的拥有各种中文译名的算法。



Edsger Wybe Dijkstra

【算法思路】

我们可以用邻接矩阵或邻接表来存储图。

还需要用一维数组(dis)来存储某一点（源点）到其余各点的当前距离，最终它会变成某一点到其余各点的最短路径长度（若两点不联通，那么它们的最短路径长度可记为正无穷）。注意初始化。

用book数组记录顶点是否已被求出与源点的最短路径。

在所有没有求出最短路径的顶点中，每次选择一个离源点最近的点，在book数组中标记它为真值，并对每一条以它为起点的边进行松弛操作，更新dis数组。

【算法运用】

Dijkstra是一种基于贪心的算法，运用在稠密图上求单源最短路。不能解决负权图，但具有良好的可扩展性，可以用它的思想解决一些有趣的问题。用堆优化后时间复杂度为O((M+N)log N)，空间复杂度为O(M)。

【正确性证明】

它的正确性是显然的。

（本来笔者想直接上代码，但还是稍微解释一下）

在非负权图中，若存在以某点u为起点的边中最短边的终点点v，则u、v最短路长度为此边的长度l。

因为如果从u再经过若干个点到达v，路程一定会比l大（非负权图，l为以u为起点的最短边长度）。

【代码实现】

这里给出Dijkstra的其中一种直接、通俗易懂的实现方法，用邻接矩阵存储。

下面请欣赏不出现ele、大括号不换行的优美代码。

D-matrix.cpp

当然，也可以用邻接表来存储，它在稀疏图上会优于用邻接矩阵的方法。