　由于此题还未上PAT官网题库，故没有测试集，仅仅是通过了样例，若发现错误，感谢留言指正。

Solution：

　　这道题不难，首先针对每个咨询，都对其出发点进行Dijkstra，然后判断询问的数组顺序是不是从小距离到大距离的，是的话那么就是Dijstra了，否则不是。

　　当然也可以在Dijstra的过程中直接判断，即每次选中的中间节点的最短距离是不是满足给出数组的那个节点，是的话继续。否则直接false

方法二 ：这道题不是“常规的Dijkstra的题（也就是像1087那种套个模板就行的）”，但是不用恐惧，只要仔细分析一下：

题目给的输入的最后一个序列 3 2 1 5 4为什么不行呢？

因为在第3个数（即1）的时候，起点3到点1的距离d[1]为3（点3->点2->点1），但是最短路径为1（点3->点4）.

也就是说给出的序列是Dijkstra序列的关键在于，序列的第i个数x的d[x]是等于当时求到的最短路径。

<https://blog.csdn.net/allisonshing/article/details/100643092> 这里有测试集

方法三

**注意审题** Orz：检查序列是否为 Dijkstra 过程中 每次找到 距离原点最近 用于更新相邻点 的 中介点 的 序列

* Dijkstra 每次找目前距原点最近的点，若有几个距离相同的会取其中一个，但其实先检查另外几个点也可
* [整体模板还是用 Dijkstra](https://www.cnblogs.com/whale90830/p/11493567.html)，每次循环检查 query 的一位
  + 查 当前距离原点最近 的 未访问点 与原点的距离
  + 看 与当前位的 query 中的点 到原点的距离是否符合
  + 若符合，则以 query 当前点为中介 更新邻接点距原点的距离
  + 若不符合，说明此时不应当以 query 当前点为中介继续 Dijkstra，即此 query 非所要求的 Dijkstra sequence，返回 false

我的算法：

Dijkstra’s Algorithm

S是一个集合， 包括了原点s和所有已经发现的路径。

走U1过路更短，那就更新distance [ u2 ] = distance [ u1 ] + length(< u1, u2>).

V = smallest unknown distance vertex;

T = O( |V|^2 + |E| )

Distance 距离数组

static vis[1001]

query[1001];查询数组

G[1001][1001] 图table

bool Dijkstra(int root){

For( 遍历每一个节点)

原点到原点的距离为0

如果原点到j的路径距离比min小,而且没有遍历过这个点, 那就更新min.

把每个点记录完了, 开始查询.

如果查询的节点query[i]到原点的距离是min,，则以 query 当前点为中介 更新邻接点距原点的距离

不符合，说明此时不应当以 query 当前点为中介继续 Dijkstra，即此 query 非所要求的 Dijkstra sequence，返回 false

更新邻接点距原点的距离的方法是:

1/这个点已经查过了. 可以用 0 和1来标记, 1 表示记过了 static vis[1001]

2/ 遍历每一个节点

并且到有这条边 &&如果他到原点距离> 他到查询节点+查询节点到原点距离,&&没有遍历过这个点&&

那就更新 这个节点到原点的距离.

/利用for循环更新节点. 时间复杂度V^2 .

改进方法: 遍历和他相连的节点,