

POJ3984

**题目描述 ( POJ3984 )**：用一个二维数组表示一个迷宫，其中 1 表示墙壁，0 表示可以走的路，只能横着走或竖着走，不能斜着走，编写程序，找出从左上角到右下角的最短路线。

```
int maze[5][5] = {
0, 1, 0, 0, 0,
0, 1, 0, 1, 0,
0, 0, 0, 0, 0,
0, 1, 1, 1, 0,
0, 0, 0, 1, 0,
};
```

**输入**：一个 5×5 的二维数组，表示一个迷宫。数据保证有唯一解。

**输出**：从左上角到右下角的最短路径，格式如以下输出样例所示。

输入样例	输出样例
0 1 0 0 0	(0, 0)
0 1 0 1 0	(1, 0)
0 0 0 0 0	(2, 0)
0 1 1 1 0	(2, 1)
0 0 0 1 0	(2, 2)
	(2, 3)
	(2, 4)
	(3, 4)
	(4, 4)

POJ3635

**题目描述 ( POJ3635 )**：城市之间的油价是不一样的，编写程序，寻找最便宜的城市间旅行方式。在旅行途中可以加满油箱。假设汽车每单位距离使用一单位燃料，从一个空油箱开始。

**输入**：输入的第 1 行包含 n ( 1≤n≤1000 ) 和 m ( 0≤m≤10 000 )，表示城市和道路的数量。下一行包含 n 个整数 p<sub>i</sub> ( 1≤p<sub>i</sub>≤100 )，其中 p<sub>i</sub>表示第 i 个城市的燃油价格。接下来的 m 行，每行都包含 3 个整数 u、v ( 0≤u, v<n ) 和 d ( 1≤d≤100 )，表示在 u 和 v 之间有一条路，长度为 d。接下来一行是查询数 q ( 1≤q≤100 )。再接下来的 q 行，每行都包含 3 个整数 c ( 1≤c≤100 )、s 和 e，其中 c 是车辆的油箱容量，s 是起点城市，e 是终点城市。

**输出**：对于每个查询，都输出给定容量的汽车从 s 到 e 的最便宜旅程的价格，如果无法从 s 到 e，则输出 “impossible”。

输入样例	输出样例
5 5	170
10 10 20 12 13	impossible
0 1 9	
0 2 8	
1 2 1	
1 3 11	
2 3 7	
2	
10 0 3	
20 1 4	