

POJ3984

迷宫问题

Time Limit: 1000MS

Memory Limit: 65536K

Total Submissions: 9903

Accepted: 5868

Description

定义一个二维数组：

```
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,
```

```
};
```

它表示一个迷宫，其中的 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，要求程序找出从左上角到右下角的最短路线。

Input

一个 5×5 的二维数组，表示一个迷宫。数据保证有唯一解。

Output

左上角到右下角的最短路径，格式如样例所示。

Sample Input

```
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
```

Sample Output

```
(0,0)
```

(1, 0)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(2, 4)
(3, 4)
(4, 4)

参考代码:

```
#include<iostream>
using namespace std;
int map[5][5];
int vis[5][5];
struct node{
    int x;
    int y;
    int pre;
}edge[100];
int front=0,rear=1;//队头，队尾
int dir[4][2]={{0,-1},{1,0},{0,1},{-1,0}};
void f(int i)//倒向追踪法
{
    if(edge[i].pre!=-1)
    {
        f(edge[i].pre);
        cout<<"("<<edge[i].x<<" , "<<edge[i].y<<" "<<endl;
    }
}
void BFS(int x,int y)
{
    edge[front].x=x;
    edge[front].y=y;
    edge[front].pre=-1;
    while(front<rear)//队列为空时终止
    {
        int u;
        for(u=0;u<4;u++)
        {
            int x=edge[front].x+dir[u][0];
            int y=edge[front].y+dir[u][1];
            if(x<0 || x>=5 || y<0 || y>=5 || vis[x][y]==1 || map[x][y]==1)
                continue;
```

```

        else
        {
            vis[x][y]=1;
            map[x][y]=1;
            edge[rear].x=x;//入队
            edge[rear].y=y;
            edge[rear].pre=front;
            rear++;
        }
        if(x==4&&y==4)
            f(front);
    }
    front++;//出队
}
}

int main()
{
    int i,j;
    for(i=0;i<5;i++)
    {
        for(j=0;j<5;j++)
        {
            cin>>map[i][j];
        }
    }
    memset(vis,0,sizeof(vis));
    cout<<"("<<"0, 0)"<<endl;
    BFS(0,0);
    cout<<"(4, 4)"<<endl;
    return 0;
}

```