迷宫问题

1 .问题描述

- 描述：一个m*n的二维表M(表中的元素为0或1,0表示不通, 1表示通。), 起始位置start(x0,y0),目标位置end(x1,y1).求start -> end 的一条路径。
- 输入: M, start, end
- 输出: start到end的路径

2. 算法思想

回溯

3. 步骤

3.1 自然语言描述

1. 将开始位置start加入路径(入栈).
2. 按顺序检查栈顶元素的相邻点，如果相邻点是1，则将该点加入路径（入栈）.
3. 如果相邻点都走不通，则回退（出栈）。
4. 循环执行2,3
5. 如果栈为空，则start->end没有一条可达路径.
6. 若栈顶元素是end则栈中的元素即为start->end的路径

3.2 伪代码描述

```
Maze(M,x0,y0,x1,y1)
{
    //  初始化栈
    traceStack = CreateStack();
    //栈顶元素入栈,并设置下次探测位置为0
    traceStack.push([x0,y0,0])
    while(not Empty(traceStack))
    {
        //获取栈顶元素，不弹出
        topElement = Peak(traceStack)
        //获取topElement元素相邻的元素,[x2,y2]
        x2,y2 = GetAvailableNeighbour(M,topElement)
        //相邻的点都不通
        if([x2,y2] 不存在)
        {
            //出栈
            traceStack.pop()
            //把此节点标为死胡同
            MarkUnAvailable(M,topElement)
            //换个方向再试探
            newTopElement = Peak(traceStack)
            newTopElement.Direction += 1
        }
        else
        {
            //加入路径
            traceStack.push([x2,y2,0])
        }
        //判断是否到达终点
        if(x2==x1 and y2=y1 )
        {
            return traceStack;
        }
    }
    //没有可达路径
    return null;
}
```