**项目说明文档**

**数据结构课程设计**

**——勇闯迷宫游戏**

作 者 姓 名： 戴仁杰

学 号： 1951650

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 分析 - 3 -](#_Toc91011474)

[1.1 背景分析 - 3 -](#_Toc91011475)

[1.2 功能分析 - 3 -](#_Toc91011476)

[2 设计 - 3 -](#_Toc91011477)

[2.1 数据结构设计（DFS示意） - 3 -](#_Toc91011478)

[2.2 类结构设计 - 3 -](#_Toc91011479)

[2.3 成员与操作设计 - 4 -](#_Toc91011480)

[2.3.1 Point类 - 4 -](#_Toc91011481)

[2.3.2 Node类 - 4 -](#_Toc91011482)

[2.3.3 Linklist类 - 4 -](#_Toc91011483)

[2.3.4 Maze类 - 5 -](#_Toc91011484)

[2.4 系统设计 - 5 -](#_Toc91011485)

[3 实现 - 6 -](#_Toc91011486)

[3.1 寻找路径功能的实现 - 6 -](#_Toc91011487)

[3.1.1 寻找路径功能流程图 - 6 -](#_Toc91011488)

[3.1.2 寻找路径功能核心代码 - 6 -](#_Toc91011489)

[3.2 路径展示功能的实现 - 7 -](#_Toc91011490)

[3.2.1 路径展示功能流程图 - 7 -](#_Toc91011491)

[3.2.2 路径展示功能核心代码 - 7 -](#_Toc91011492)

[3.2.3 路径展示功能截屏示例 - 8 -](#_Toc91011493)

[3.3 总体功能的实现 - 8 -](#_Toc91011494)

[3.3.1 总体功能流程图 - 8 -](#_Toc91011495)

[3.3.2 总体功能核心代码 - 9 -](#_Toc91011496)

[3.3.3 总体功能截屏示例 - 9 -](#_Toc91011497)

[4 测试 - 9 -](#_Toc91011498)

[4.1 功能测试 - 9 -](#_Toc91011499)

[4.1.1 联通迷宫寻路功能测试 - 9 -](#_Toc91011500)

[4.1.2 不联通迷宫寻路功能测试 - 10 -](#_Toc91011501)

1 分析

1.1 背景分析

相信每个人都或多或少的接触过迷宫类型的游戏，在迷宫里有着两个门，分别为入口和出口，人物需要从入口出发，避开障碍，分辨岔路口，最后找到一条路能够通往出口。本来就来设计一个程序来复刻这一类经典的迷宫游戏。

1.2 功能分析

首先一个迷宫游戏必须要有地图，这里为了让游戏更有可玩性，设置了一张6\*6大小的地图，内部设有一个出口和一个入口。符号表示方面，入口为S，出口为E，#为障碍物，0为通路。其次，要能够使游戏人物寻找通路以到达出口。最后，游戏还要能保存人物最终通过迷宫的路径并显示。

综上所述，一个迷宫游戏需要有建立地图，人物寻找通路，输出通过路径并显示的功能。

2 设计

2.1 数据结构设计（DFS示意）

如上功能分析所述，地图已经固定，使用一个二维数组存储即可。而路径由于在进行迷宫搜索时会动态变化，无法确定长度，因此决定使用Linklist链表来保存路径。迷宫最重要的是搜索通路功能，图形搜索算法最广泛使用的是深度优先搜索（DFS）与广度优先搜索（BFS），本游戏最终采用了深度优先搜索（DFS）算法。

迷宫问题的求解过程可以采用回溯法即在一定的约束条件下试探地搜索前进，若前进中受阻，则及时回头纠正错误另择通路继续搜索的方法。从入口出发，按某一方向向前探索，若能走通，即某处可达，则到达新点，否则探索下一个方向；若所有的方向均没有通路，则沿原路返回前一点，换下一个方向再继续试探，直到所有可能的道路都探索到，或找到一条通路，或无路可走又返回入口点。在求解过程中，为了保证在达到某一个点后不能向前继续行走时，能正确返回前一个以便从下一个方向向前试探，则需要在试探过程中保存所能够达到的每个点的下标以及该点前进的方向，当找到出口时试探过程就结束了。

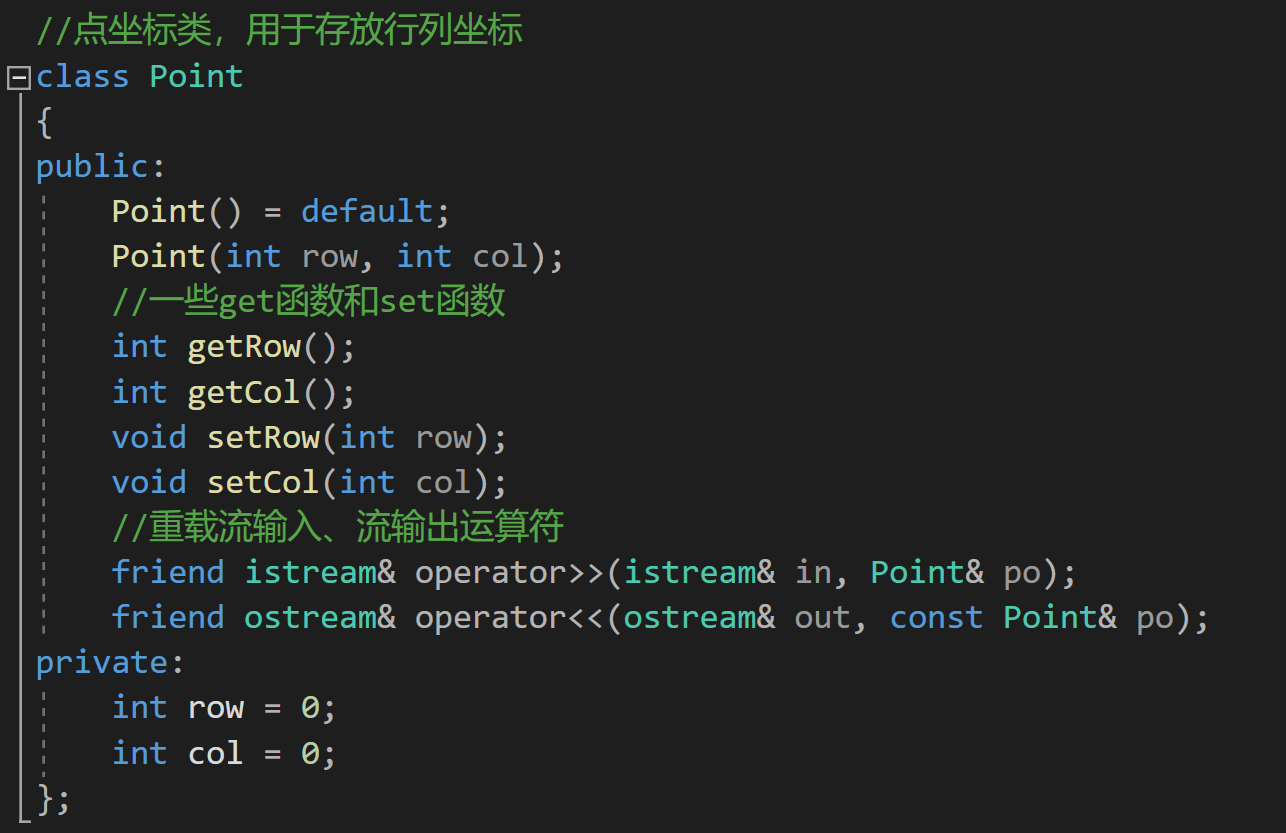
2.2 类结构设计

由于迷宫中的坐标都是点的形式，所以首先设计了一个坐标点类（Point类）便于表示点的位置。然后设计一个保存迷宫路径的链表类（Linklist），并配有其对应的链表节点类（Node）。同时为了使链表更具泛用性，将Linklist与Node设计为模板类。最后，为了更方便管理，将迷宫地图、迷宫搜索任意时刻的状态与搜索时需要的操作整合进一个迷宫类（Maze类）。

2.3 成员与操作设计

### 2.3.1 Point类

Point类存放点的横纵坐标信息，提供带参构造函数，并重载了流输入输出运算符用于外接直接输出其内容。为了保持封装性，数据设为私有，并提供了一系列get与set函数与外界进行数据交流。



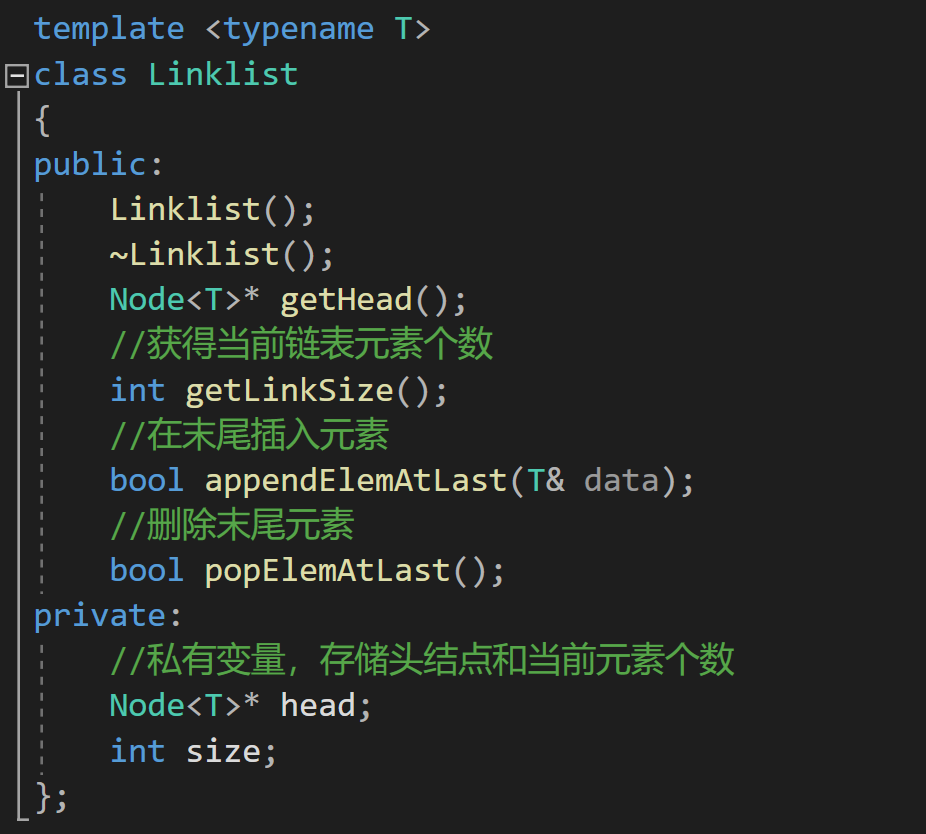
### 2.3.2 Node类

Node类为链表节点类，采用了类模板提高泛用性。提供默认构造函数、带参构造函数以及与外接数据交流的一系列get与set函数。类内数据为data数据域以及指针域，存储点的信息以及与后继节点相连。



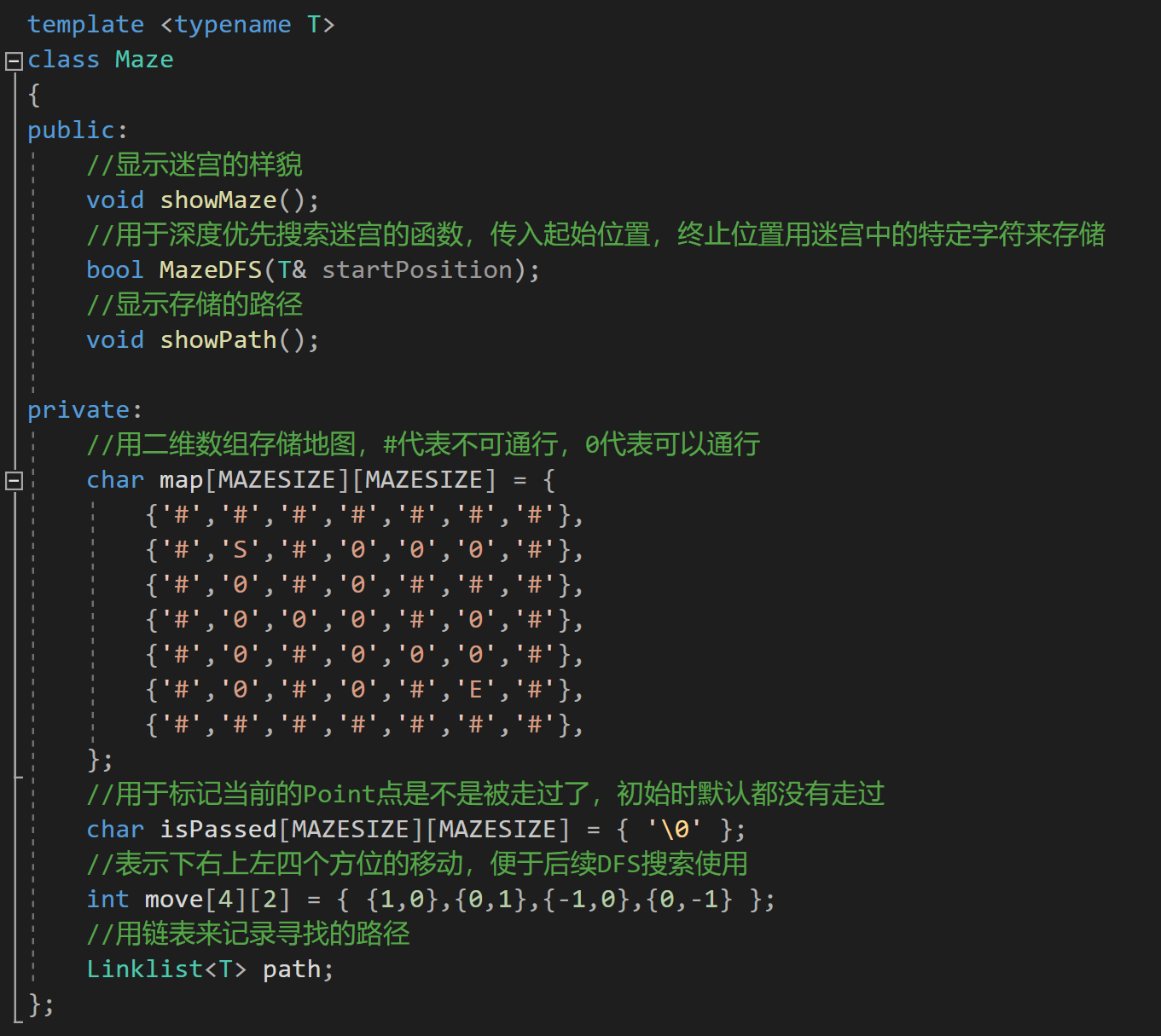
### 2.3.3 Linklist类

Linklist类为链表类，采用了类模板提高泛用性。提供构造函数新建链表，析构函数释放内存。提供get系列函数获得链表头指针以及链表大小，并提供尾部插入和尾部删除函数供外界调用。类内数据为表头指针和链表大小。



### 2.3.4 Maze类

Maza类为迷宫类，采用了类模板提高泛用性。迷宫类封装了展示迷宫、DFS算法搜索迷宫以及显示路径的方法。存储方面，Maze类有一个二维数组map作为地图，并有一个一维数组isPassed记录点是否已经走过，用path链表记录搜索的路径。此外为了移动方便，还设计了一个move数组存储四个方位的移动信息。



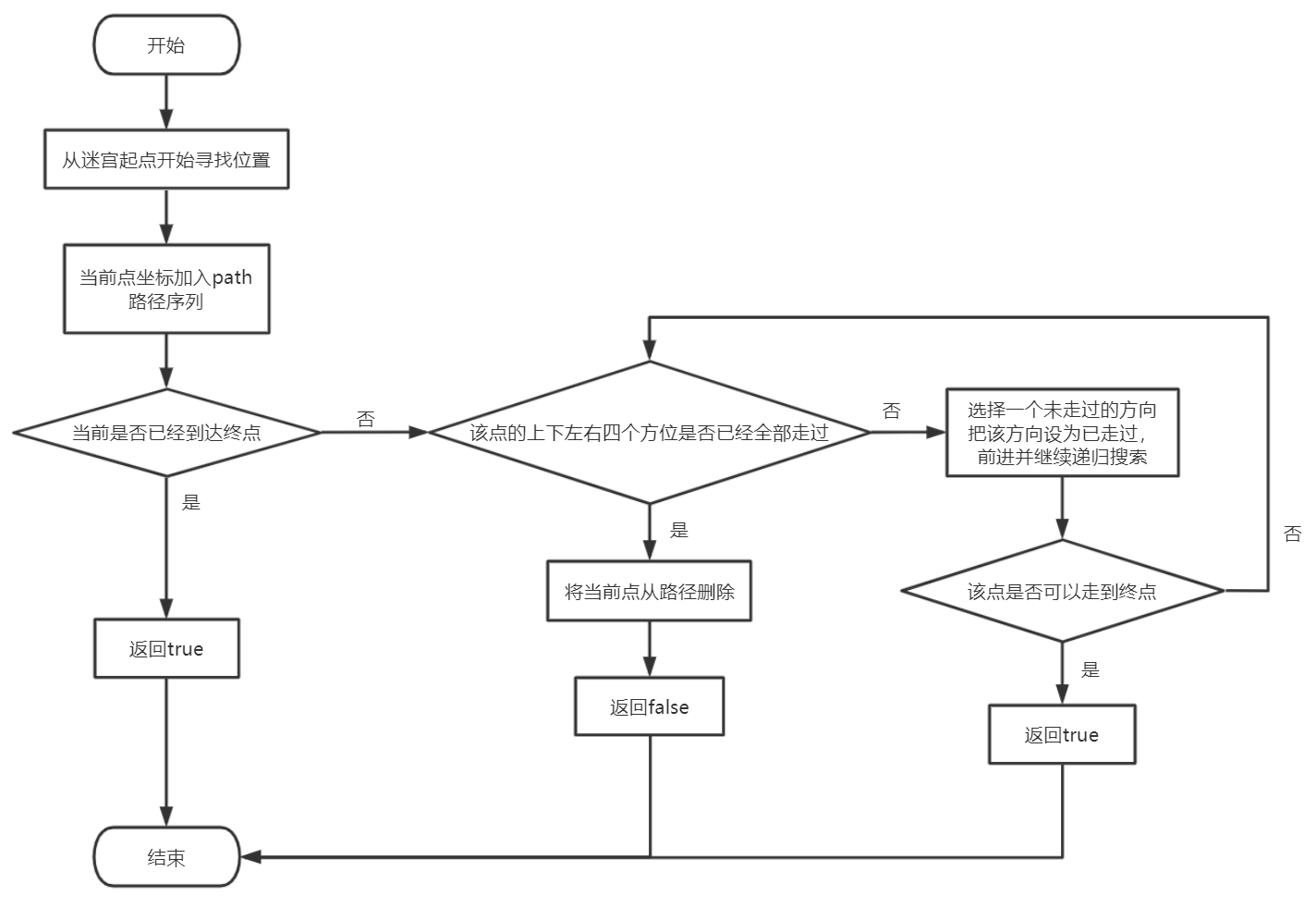
2.4 系统设计

开始游戏后，首先初始化的地图并显示，其中S代表起点，E代表终点，0代表可以走的点，#代表不可以走的点。随后游戏会搜索通路路径，完毕后提示用户按下任意键继续，如果没有通路则会输出提示，如果有通路，则会输出在迷宫中的通路，以及从起点到终点，路径中每一个点的坐标。

3 实现

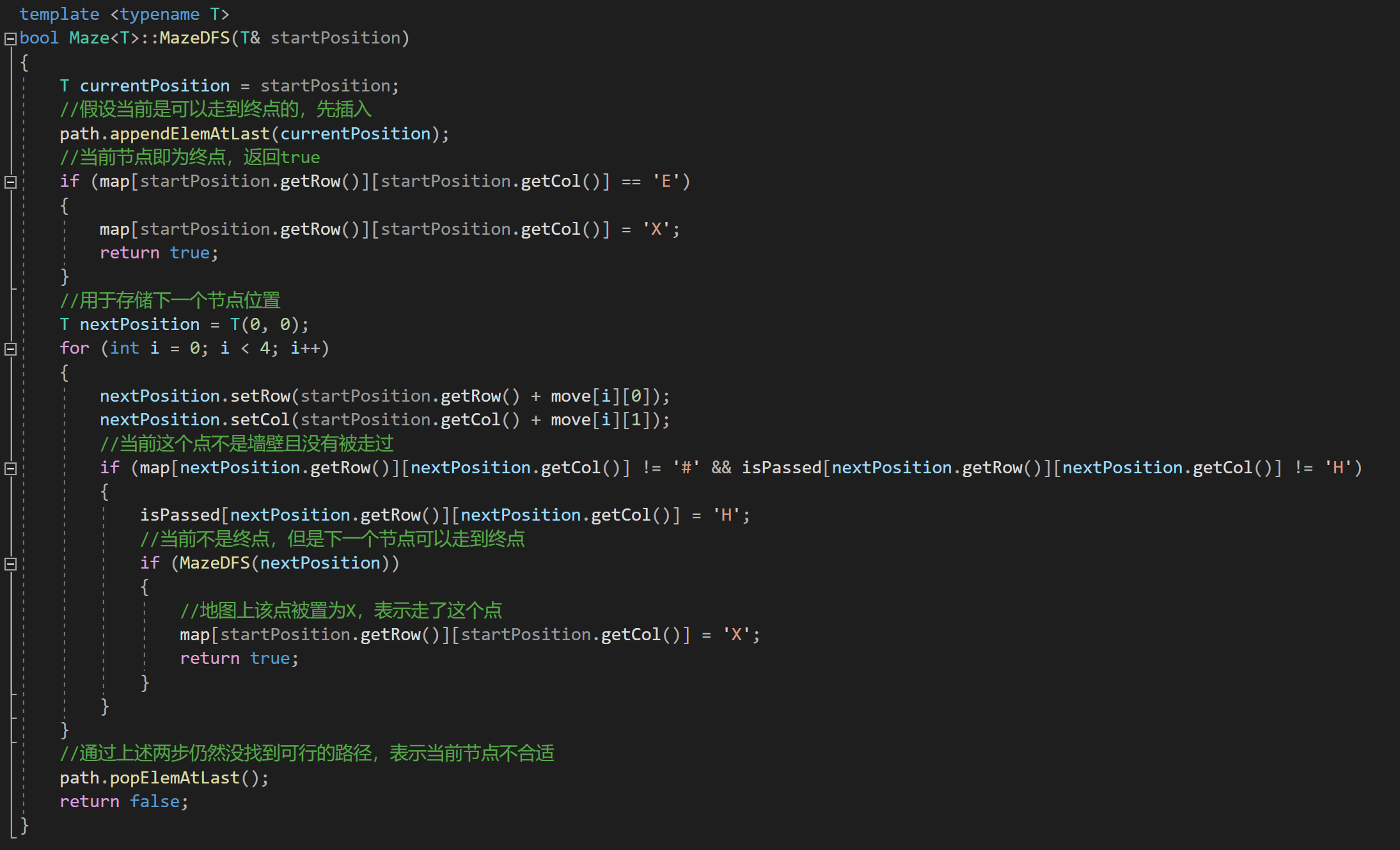
3.1 寻找路径功能的实现

3.1.1 寻找路径功能流程图



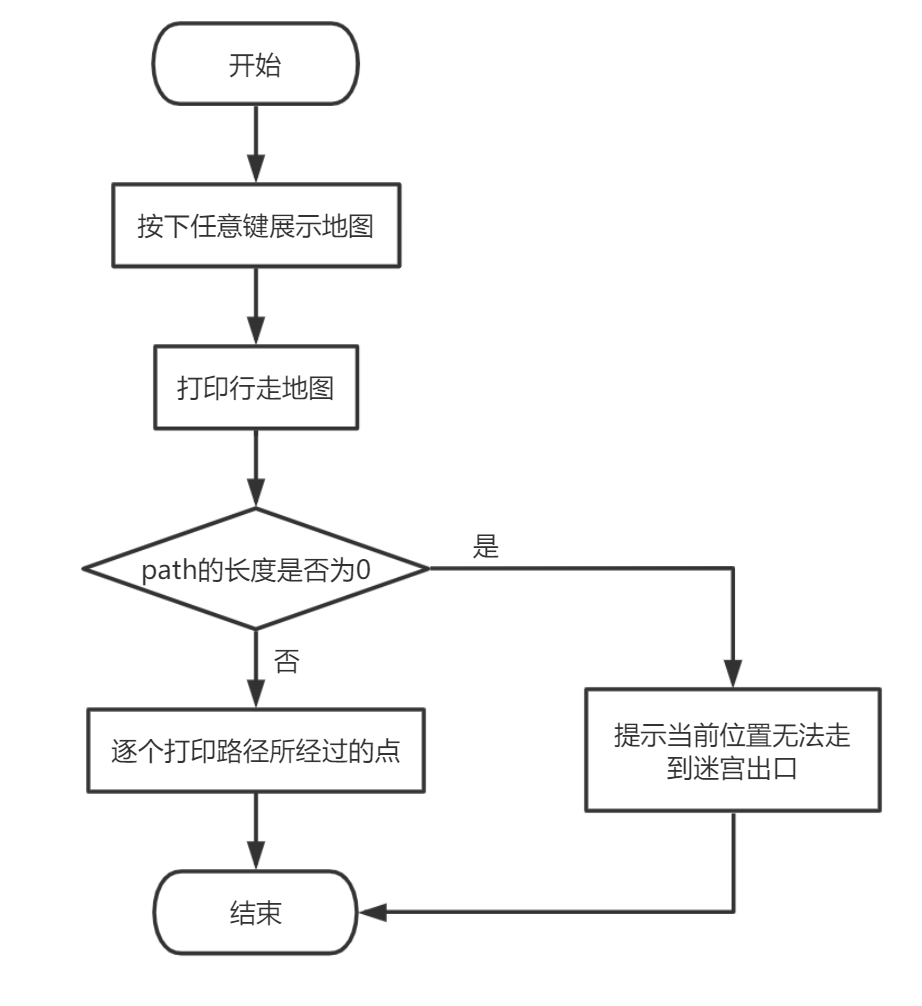
3.1.2 寻找路径功能核心代码

1. Maze类的MazeDFS(T& startPosition)函数



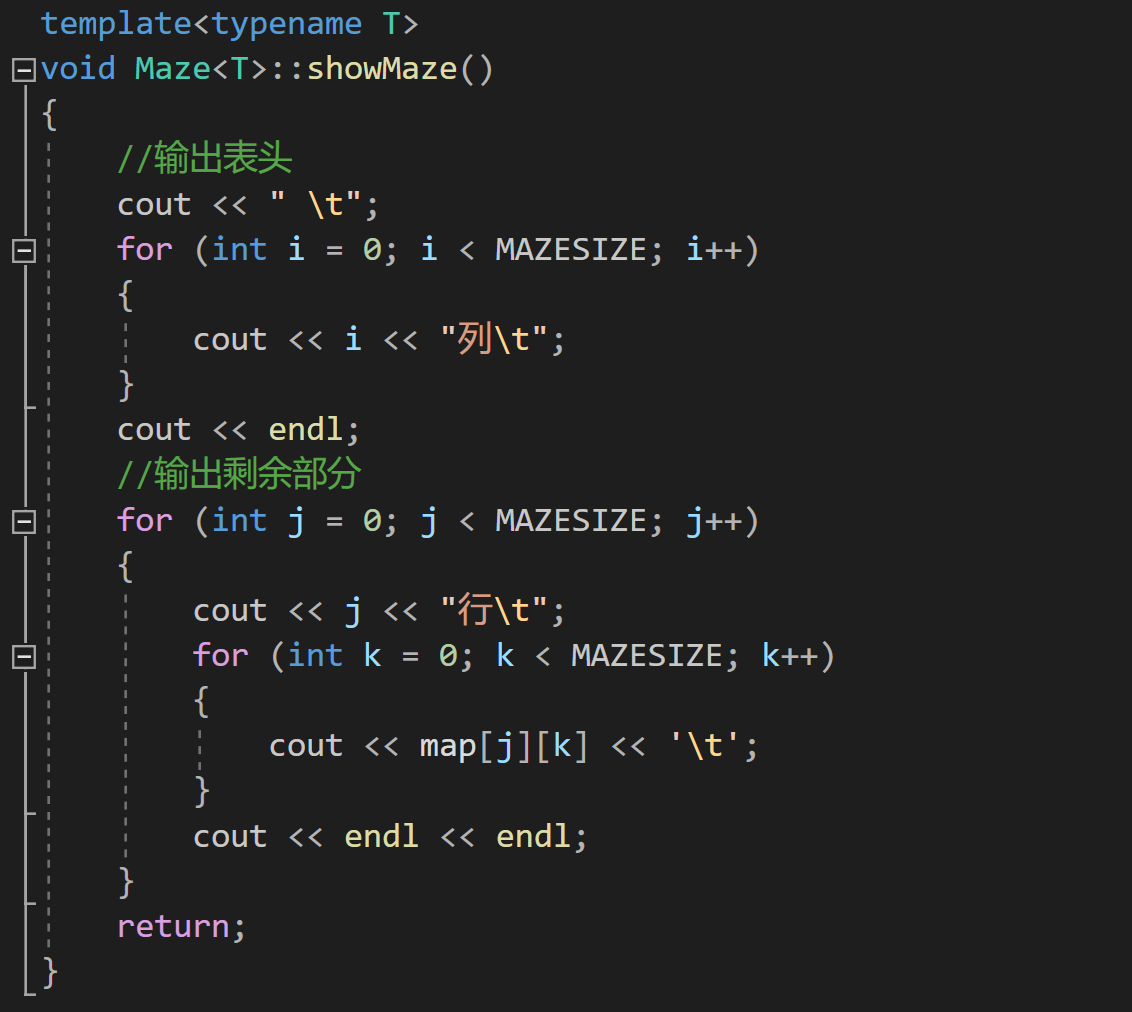
3.2 路径展示功能的实现

3.2.1 路径展示功能流程图



3.2.2 路径展示功能核心代码

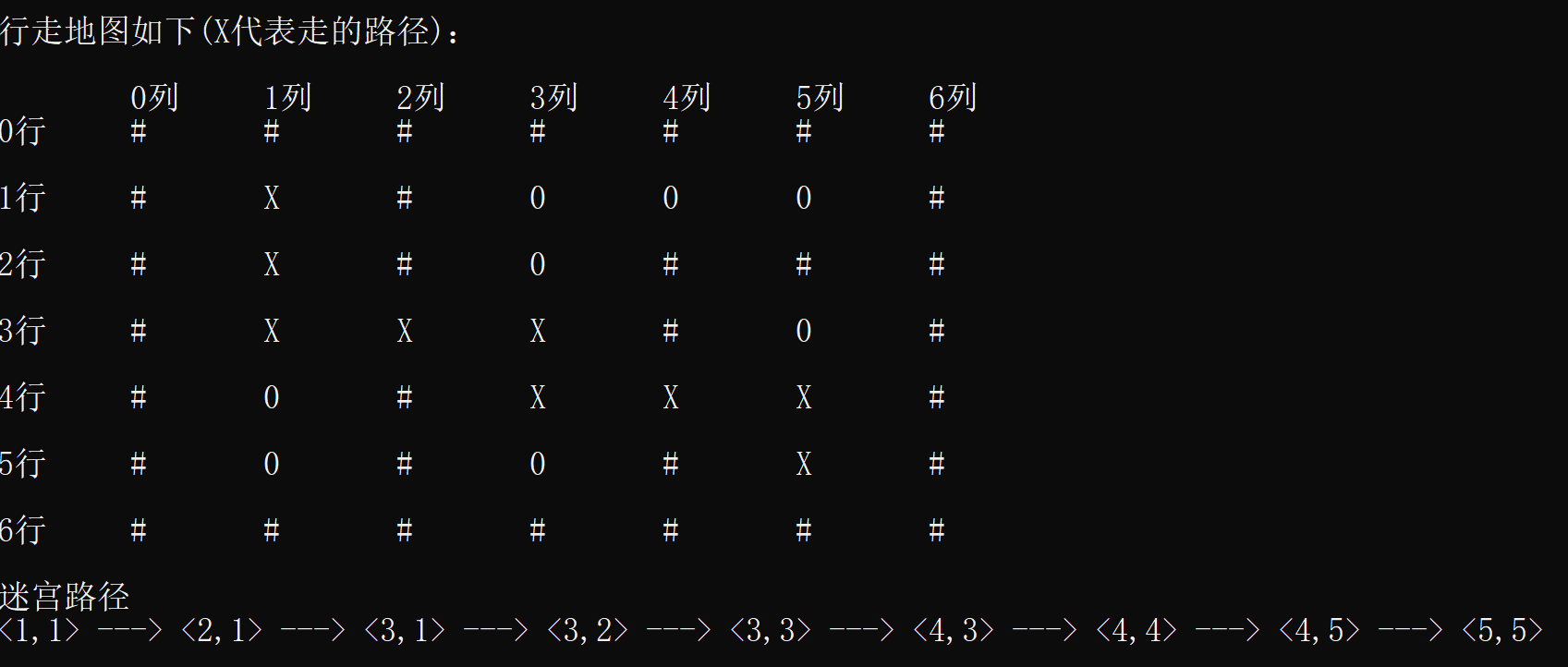
1. Maze类的showMaze()函数



2. Maze类的showPath()函数

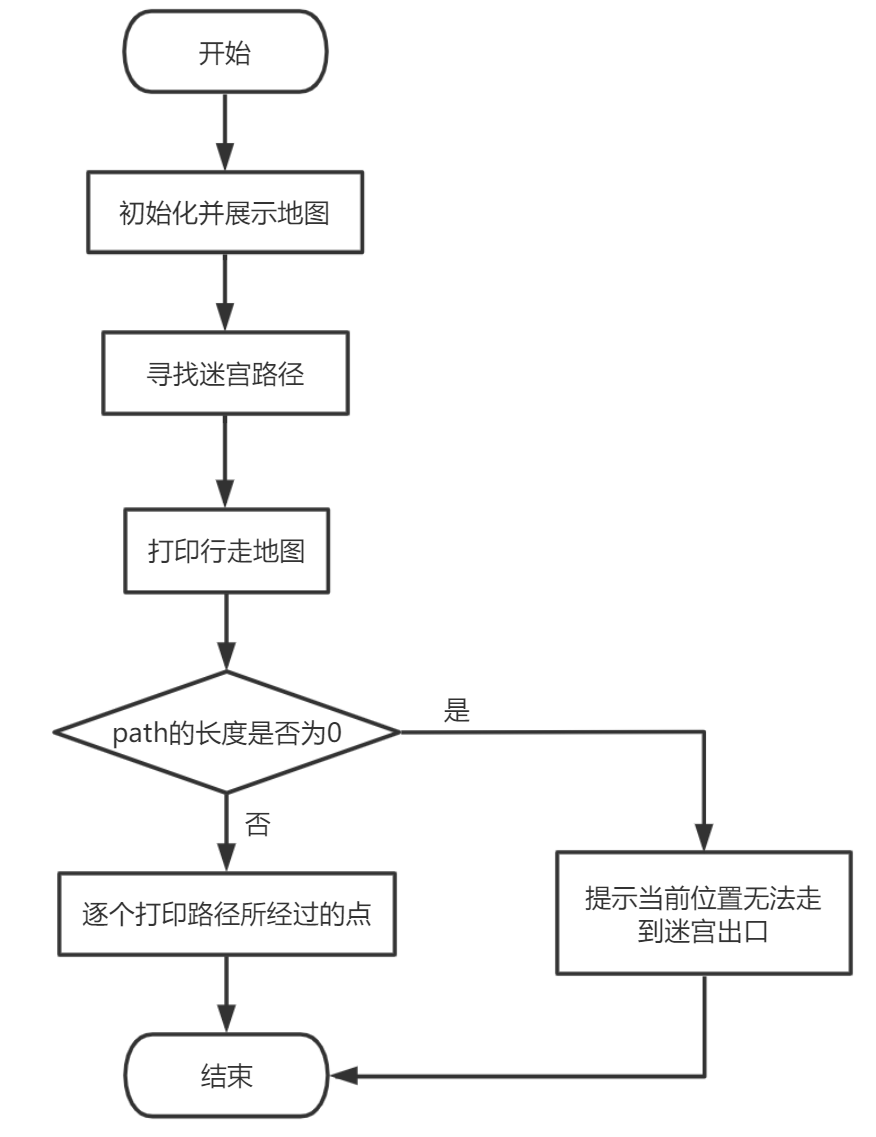


3.2.3 路径展示功能截屏示例

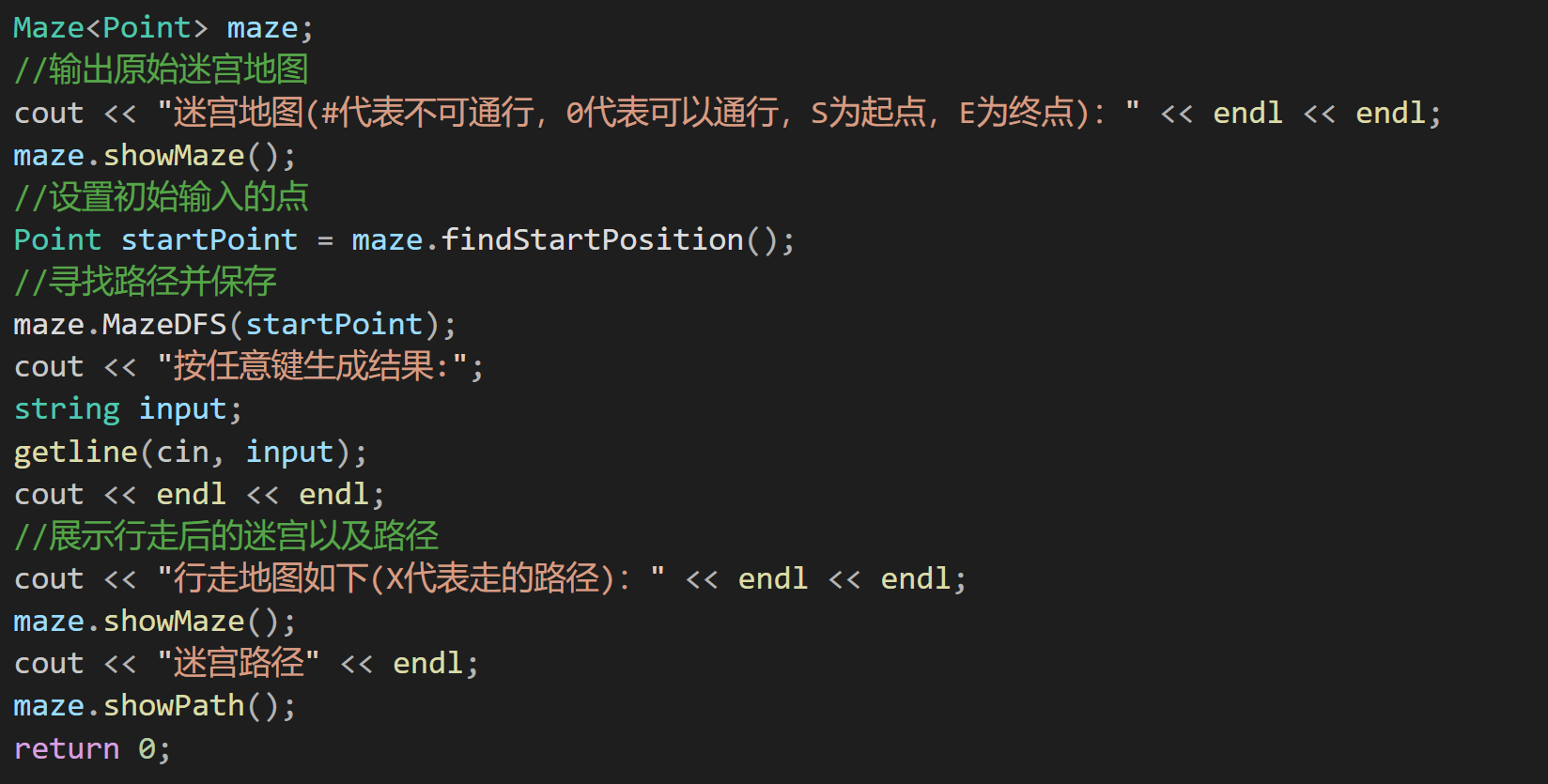


## 3.3 总体功能的实现

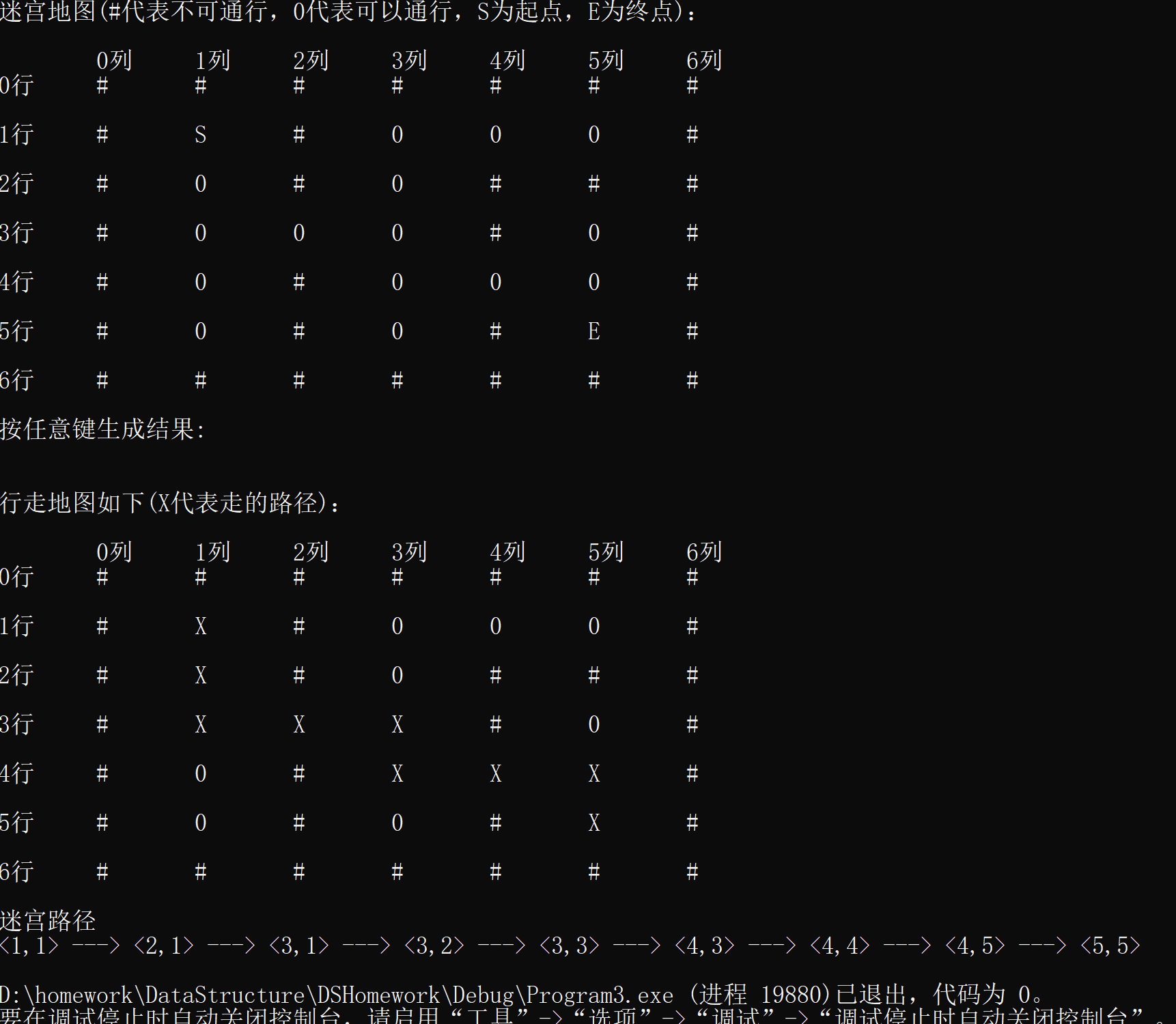
### 3.3.1 总体功能流程图



### 3.3.2 总体功能核心代码



3.3.3 总体功能截屏示例



4 测试

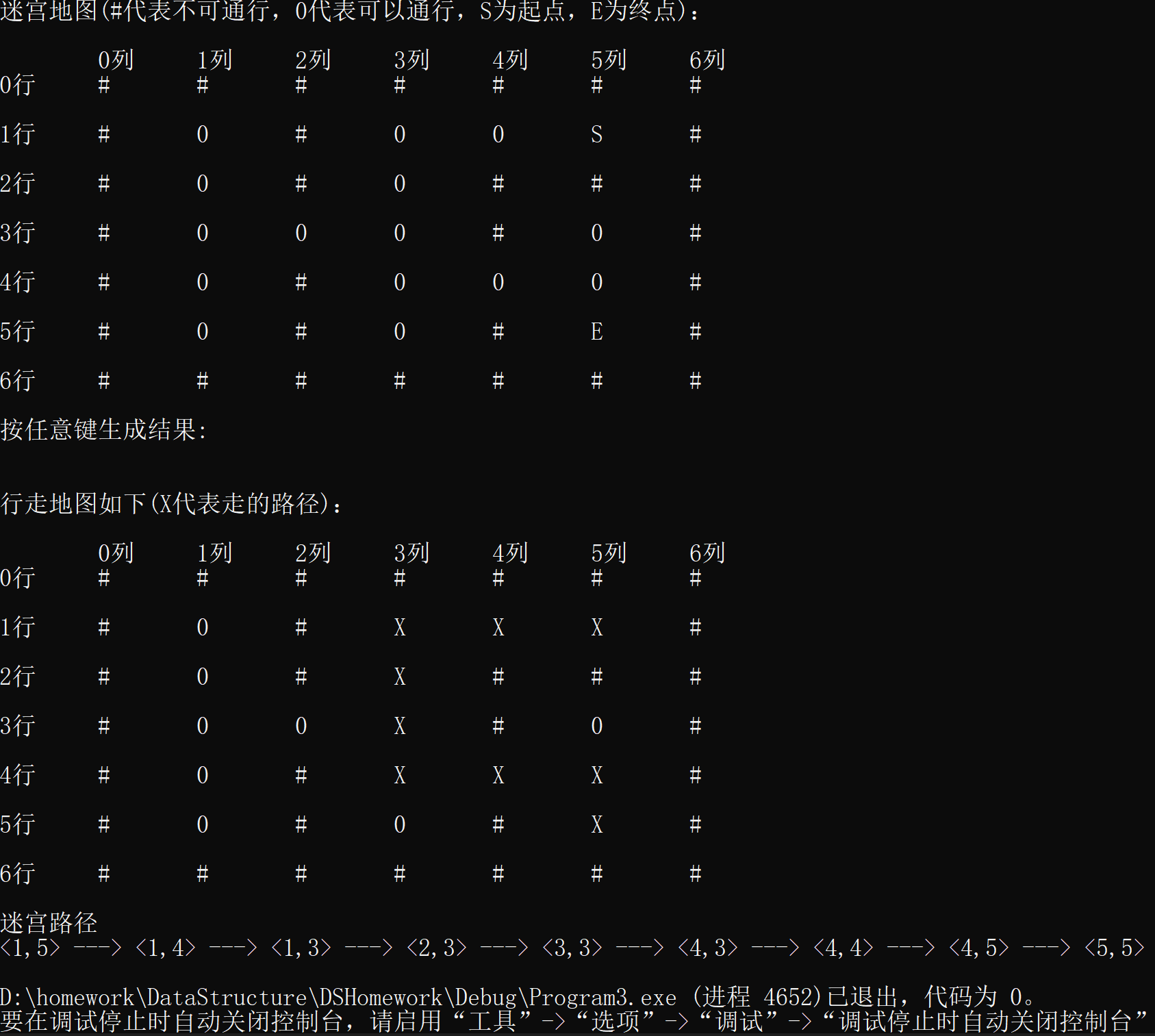
4.1 功能测试

4.1.1 联通迷宫寻路功能测试

**测试用例**：一个可以联通的地图，(1,5)起始，(5,5)终点

**预期结果**：正常寻路并给出结果

**实验结果：**



4.1.2 不联通迷宫寻路功能测试

**测试用例：**一个不可联通的地图（Demo地图在(3,2)点被堵住）

**预期结果**：输出迷宫不通的提示，程序不崩溃

**实验结果：**

