**项目说明文档**

**数据结构课程设计**

**——勇闯迷宫游戏**

作者姓名：香宁雨

学号：1954098

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.功能分析 3](#_Toc58589645)

[2.设计 4](#_Toc58589646)

[2.1数据结构设计 4](#_Toc58589647)

[2.2类结构设计 4](#_Toc58589648)

[2.3类的成员和方法 4](#_Toc58589649)

[2.4测试函数 5](#_Toc58589650)

[3.实现 6](#_Toc58589651)

[3.1SetFixedMaze()方法 6](#_Toc58589652)

[3.2SetMaze()、SetEntrance()、SetExit()方法 7](#_Toc58589653)

[3.3Dfs(int x ,int y)方法 9](#_Toc58589654)

[3.4PrintMaze()方法 10](#_Toc58589655)

[3.5PrintAnswer()方法 11](#_Toc58589656)

[4.输出测试 12](#_Toc58589657)

[4.1固定迷宫输出 12](#_Toc58589658)

[4.2手动输入迷宫输出 12](#_Toc58589659)

# 1.功能分析

我们现在有一个迷宫，迷宫只有两个门，一个门叫入口，另一个门叫出口。一个骑士骑马从入口进入迷宫，迷宫设置很多障碍，骑士需要在迷宫中寻找通路以到达出口。

题目会输入一个迷宫地图并让我们输出迷宫从入口到出口的迷宫路径，默认(1,1)为入口地址，(n-2,n-2)为出口地址。

迷宫问题的求解过程可以采用回溯法即深度优先搜索(DFS)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.设计

## 2.1数据结构设计

本次题目输入为迷宫，是一个m\*n的地图，采用数组进行处理便于遍历图的每一个方向并作出判断，还可以直观的打印出当前地图使得数据处理变得更加直观，所以本次我们采用二维数组这种连续的存储结构进行存储。

## 2.2类结构设计

因为本次的数据结构较为简单，所以本次类的设计将会在数组的基础上进行。

根据题目描述，我们的所有操作都是对同一个迷宫类对象进行的，所以我们设置一个迷宫类，并给他添加我们所需的方法和成员。为实现良好的封装性，我们将所需要的操作都封装为迷宫类对象的方法，通过调用方法来实现题目要求。

## 2.3类的成员和方法

2.3.1迷宫类(Maze)



## 2.4测试函数

本项目中采用solve()函数作为测试函数来验证输出迷宫路径的正确性，其中包括用户输入提示和错误输入报错，大大增加了程序的用户友好性和健壮性。同时，因为solve()函数是独立于main()的独立函数，所以更好的体现了面向对象语言的特性，使得外部程序可以直接调用函数接口来实现程序内容。

# 3.实现

## 3.1SetFixedMaze()方法

在本项目中，我们有两种迷宫的输入方法，一种为固定迷宫输入方法，另一个为手动输入迷宫方法，SetFixedMaze()为固定迷宫输入方法，可以进行固定迷宫即题目中所演示的迷宫的输入并输出迷宫路径。



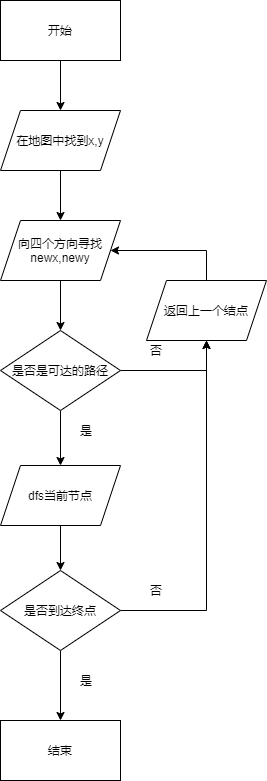
## 3.2SetMaze()、SetEntrance()、SetExit()方法

SetMaze()方法、SetEntrance()方法、SetExit()方法共同构成了手动输入迷宫的方法，通过设置迷宫、设置起点以及设置终点使程序获得迷宫的完整信息，进而得出结论。



## 3.3Dfs(int x ,int y)方法

#### 3.3.1流程图



#### 3.3.2代码实现



## 3.4PrintMaze()方法

我们通过PrintMaze()方法来可视化当前输入的迷宫或固定迷宫，让迷宫得以被用户看到，增加了用户友好性。



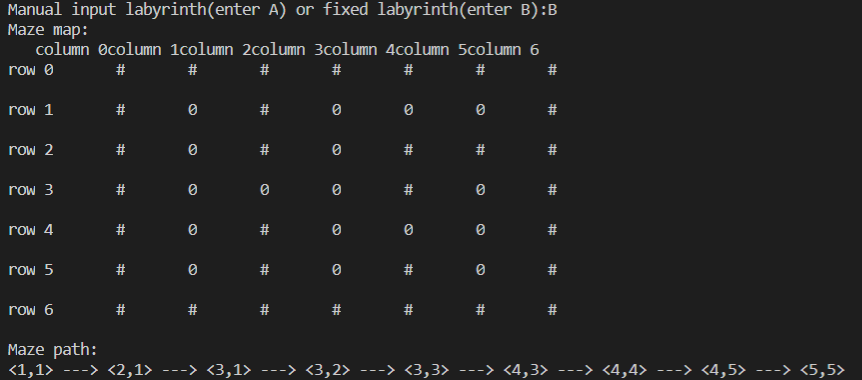
## 3.5PrintAnswer()方法

通过调用PrintAnswer()方法将我们获得的答案输出。



# 4.输出测试

## 4.1固定迷宫输出

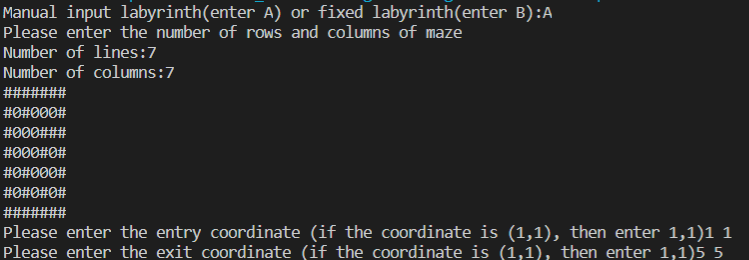


## 4.2手动输入迷宫输出

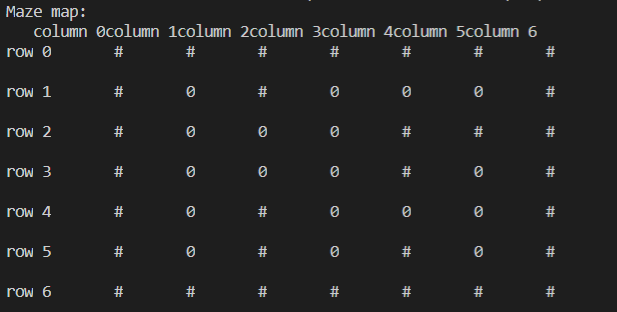
为了验证当前代码特性，我们采用一个具有多条路径答案的迷宫进行输入。本迷宫为7\*7迷宫，以（1，1）为起点，（5，5）为终点，迷宫如下：

#######  
#0#000#  
#000###  
#000#0#  
#0#000#  
#0#0#0#  
#######

设置迷宫初始化条件：



迷宫可视化：



路径输出：

