勇闯迷宫游戏

#### 软件工程20级2班 2053300 胡锦晖

# 项目概述

## 项目背景

现存在一个矩形迷宫，只有两个门与外界连通，分别为迷宫的入口与出口。迷宫中存在很多障碍，只有特定的位置可供行走。现有一名骑士从入口处进入迷宫，他需要在迷宫中寻找同路以到达出口。

## 项目目标

计算骑士从入口抵达出口的路径，并绘制路线图。

# 实现思路

## 2.1 数据的存储结构

由于依照项目示例描述，该迷宫为一个矩形迷宫，所以考虑用一个char型的二维数组作为矩阵MazeMap来存储迷宫，Mazemap[i][j]中下标i，j分别代表该位置在迷宫中的行号和列号，当该位置值为#时，表示此处为障碍；为0时表示此处为通路。

由于采用深度优先搜索的方法，还需要一个与Mazemap数组等大的bool型数组visited存储每个位置是否已经被访问过的信息，true：已被访问过；false：尚且未被访问过。

最终的路线不再另外开辟存储空间进行存储，而是在计算路线的时候进行同步输出。

## 2.2 功能实现

### 2.2.1 基本思路

可以参考Depth First Search的思想，从迷宫起点开始，在不遇到障碍或迷宫边界且当前位置未曾被搜索到的情况下，搜索当前位置的前后左右四个方向，直到某次搜索的位置为迷宫出口结束。

### 2.2.2 路线输出

将递归实现的深度优先搜索函数的返回值类型改为bool类型，只有在最终递归到“出口”（实际上是问题的入口，后面会提到原因）位置时，函数返回true，然后根据函数栈帧层层回退，依次格式化输出当前所在位置，同时将当前位置在地图数组中标记为\*，最终就可以得到一个完整的路线及标记了路线的迷宫地图。

还有一个细节需要注意：由于深度优先搜索函数的递归顺序问题，我们需要调换入口和出口的位置，即从出口开始搜索直到找到入口时返回true，然后根据递归的顺序输出得到的才是真正由入口到出口的路线。

### 2.2.3 地图输入

根据题目要求，本项目提供了两种迷宫地图的获取方式：

1.直接使用内置6\*6的示例地图：

屏幕上有字

描述已自动生成选择方式：在提示是否使用内置地图的时候键入1即可。

即入口坐标为(0, 3)，出口坐标为(5, 3)

2.使用用户自行输入的地图：

选择方式：在提示是否使用内置地图的时候键入除去数字1之外的其他任何值均可。

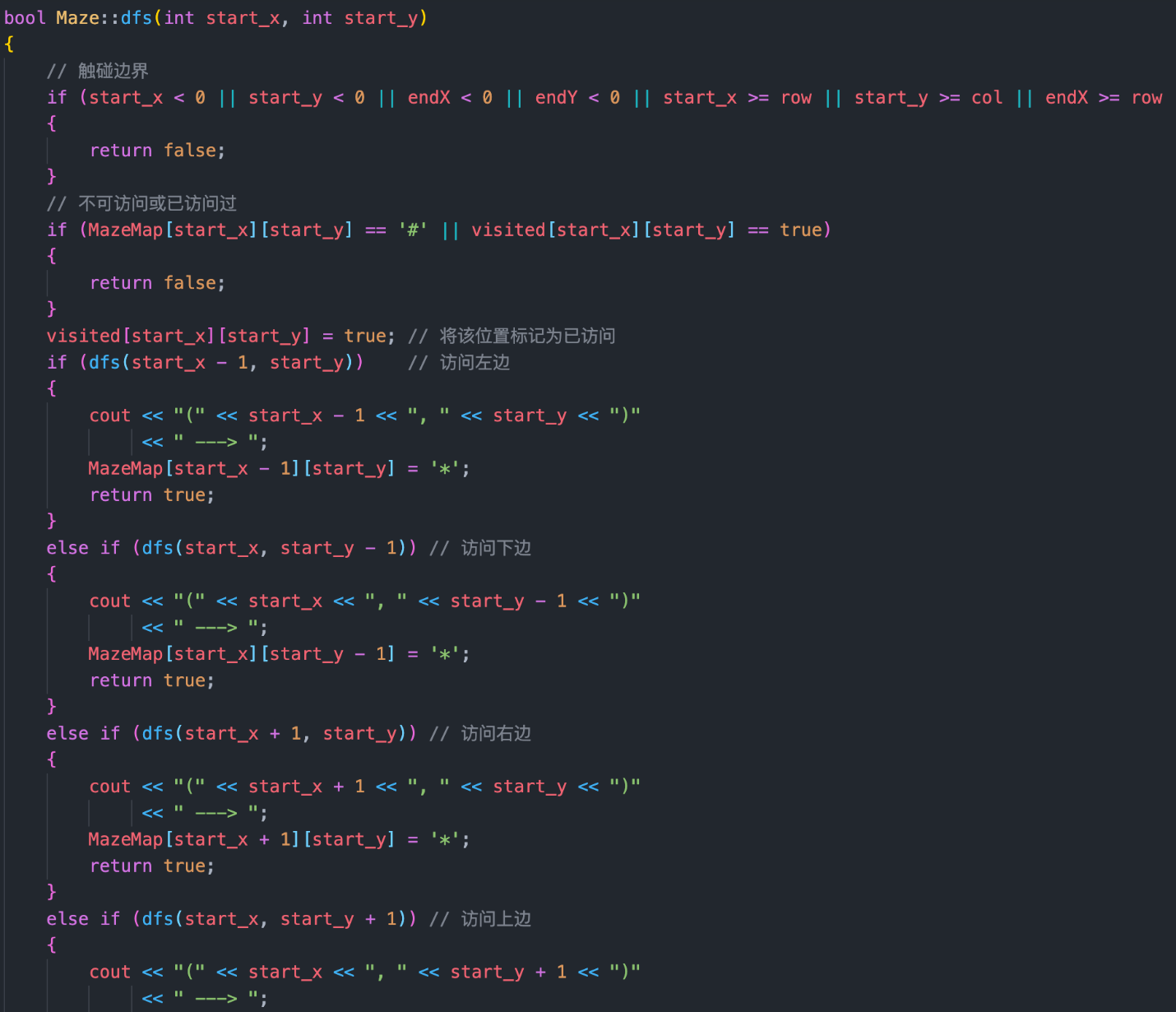
自行输入地图时需要注意的是：首先输入矩形迷宫的行数和列数，然后按照行数由小到大的顺序输入每一行的数据（均由0或#组成），且在每一行结束之后需要键入回车以换行，否则会造成不可预期的错误。此外，在输入迷宫入口和出口坐标时必须选择可以访问的位置（即值为0的位置），否则程序会提示输入错误。

注明：本项目中的所有坐标表示都是以左上角的点(0, 0)原点，右方向（行）为x轴正向，下方向（列）为y轴正向。

## 2.3 核心算法及流程图

## 图示 描述已自动生成2.4 核心代码

文本

中度可信度描述已自动生成

# 三、性能分析

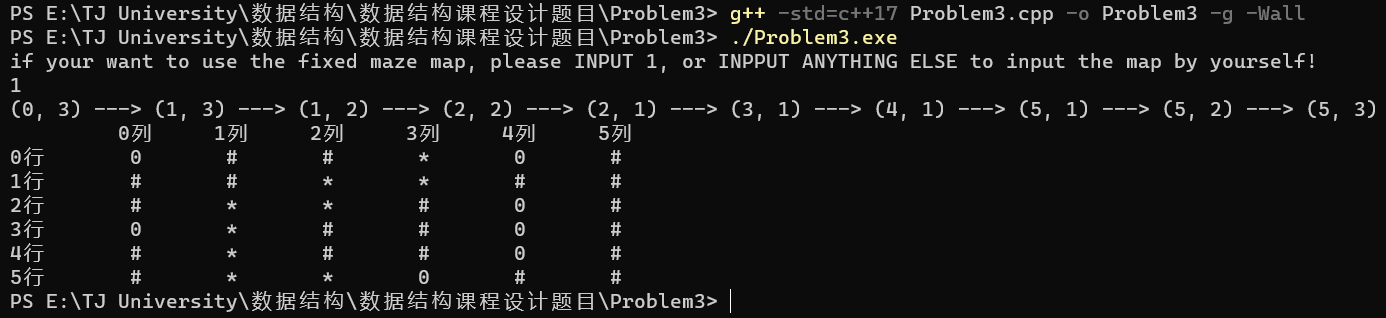
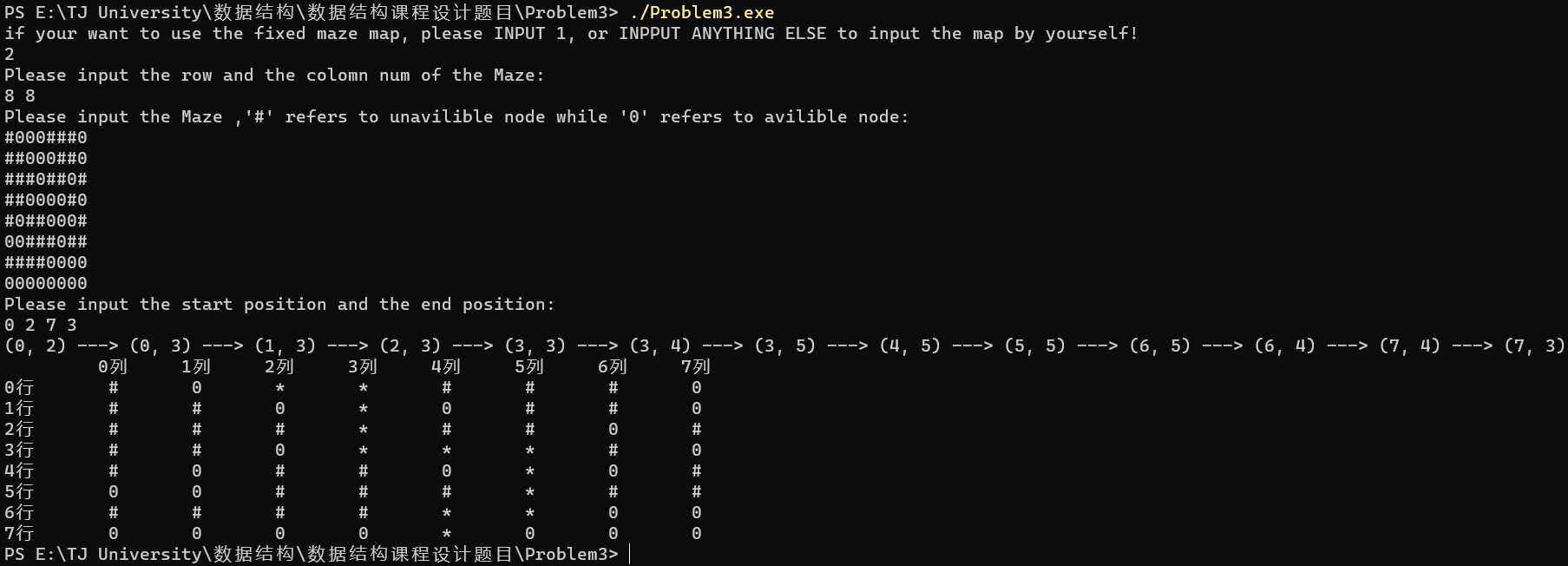
由于本题采用的是通过递归实现的深度优先搜索算法，性能来讲肯定不算优秀，而且本题由于矩形迷宫结构的特点，需要搜索上下左右四个方向，递归的规模相比一般的单级递归还要更大，整个程序执行时间有很大一部分消耗在递归时函数现场的保存以及复现上，分析渐进时间复杂度的意义也就不大了。另外，在问题规模较大的时候，还要当心堆栈溢出的情况。

# 四、测试结果

## 4.1 Windows平台

操作系统：Windows11 专业版 21H2

编译器：g++ (GCC) 8.1.0

C++标准：c++17

## 4.2 Linux平台

操作系统：CentOS Linux release 8.5.2111

编译器：g++ (GCC) 8.5.0-4

C++标准：c++17

## 4.3 Mac平台

操作系统：macOS Monterey 12.0.1

编译器：clang++ 13.0.0

文本

描述已自动生成C++标准：std=c++17

图形用户界面, 文本

描述已自动生成