项目说明文档

数据结构课程设计

——勇闯迷宫游戏

作 者 姓 名: 沈星宇

学　　号: 1951576

指 导 教 师: 张颖

学院、 专业: 软件学院 软件工程

同济大学

Tongji University

一、 分析

（1） 应用背景

迷宫问题能衍生出很多变式，单单在迷宫问题上面还可以要求寻找最短路径等等，但是每次都用两个循环嵌套去遍历的时间复杂度过高，本项目旨在用深度优先搜索算法实现从起点到终点的所有路径搜索。

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。

（2） 项目功能要求

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。

二、 设计

（1） 数据结构设计

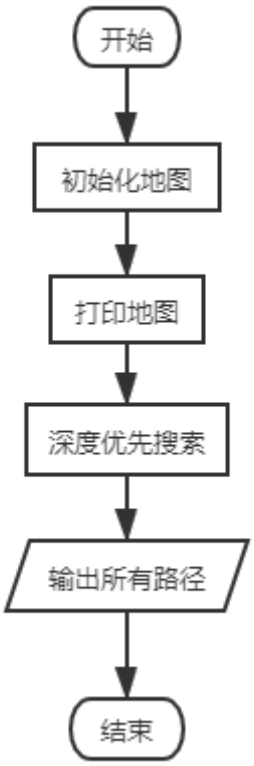
```
class Map {
private:
    char** map; //存储地图
    int** book; //标记走过的路径
    int rLength;//地图的长
    int cLength;//地图的宽
    int startX; //起点横坐标
    int startY; //起点纵坐标
    int destX;  //终点横坐标
    int destY;  //终点纵坐标
    int length; //行走的步数
    int* resultX;
    int* resultY;
    bool flag;
public:
    Map(int command,int startX,int startY,int destX,int destY);
    ~Map();
    void printMap();
    void dfs(int x, int y);
    bool getFlag();
};
```

1、 Map 类当中的私有成员存储地图的心底、起点终点信息以及最后的输出路径的信息。

2、 主要的函数有打印地图的 printMap()函数和深度优先搜索 dfs()函数

3、 本项目其实是可以实现用户输入地图进行搜索的，因为在 Map 类当中声明的是指针类型的 map 和 book，在初始化的时候也是根据长宽来动态分配内存的，但是考虑到对用户输入合法性的检测在前两个项目当中已经较好实现了，故在此项目中不再重复，直接在项目中内置了三个地图

（2） 程序流程设计



1、程序开始的时候会对程序内置的三幅地图进行初始化，并且通过printMap()函数将它打印出来

2、调用 dfs()递归函数，输出所有路径，程序结束

三、 实现

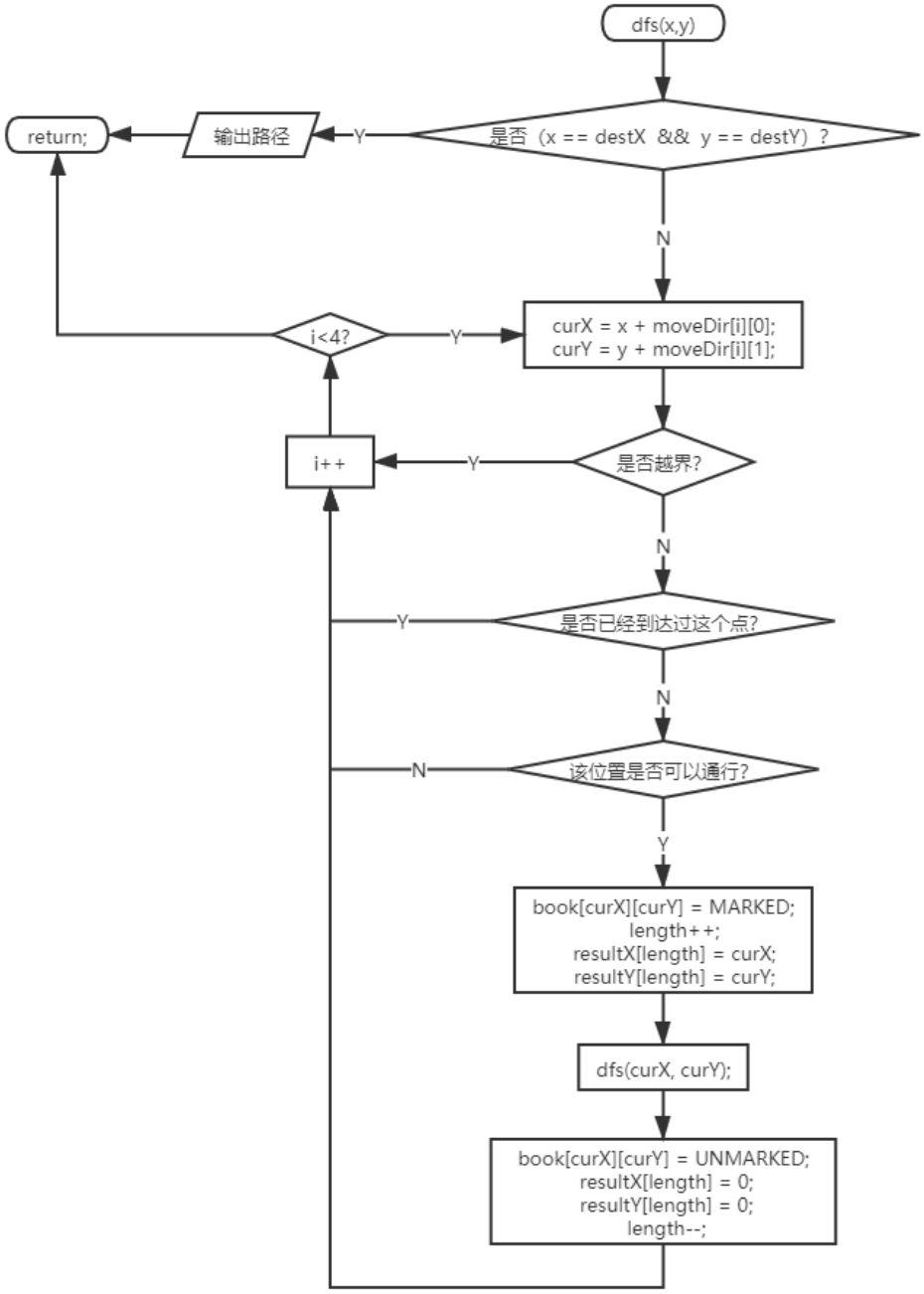
1、 main()函数的内部逻辑

（1） main()函数会首先用构造函数对 map1 进行初始化，然后调用 printMap()将其打印出来，然后调用 dfs()递归函数，传入起点的坐标

（2） 最后通过 map1 当中的 flag 的值判断是否能够找到路径到达终点

```
int startX = 1, startY = 1, destX = 5, destY = 5;
std::cout << "本程序内置了三个地图，下面是第一个地图: \n";
Map map1(COMMAND_1,startX,startY,destX,destY);
std::cout << "迷宫1地图: \n";
map1.printMap();
std::cout << "\n\n";
std::cout << "迷宫1路径: \n";
map1.dfs(startX, startY);
if (!map1.getFlag()) {
    std::cout << "没有这样的路径\n";
}
std::cout << "\n------------------------------------------------------------\n";
```

2、 dfs()的实现



(1) 递归搜索结束的条件是到达终点，即判定 x == destX && y == destY 条件，如果满足到达了终点，则输出路径，并返回上一层；如果没有到达终点，那么尝试上下左右移动（moveDir 数组分别存储了四个移动方向）

(2) 判断是否越界，是否是往回走，新的点是否可以通行等条件，我们的目标是在不越界、不往回走、新的点可以通行的条件下进行移动，在不满足任何一个条件的时候则更换移动的方向

(3) 存储当前的位置信息，将 book[curX][curY]标记为已经走过，使路径的长度增加

(4) 调用递归 dfs(curX,curY)进行下一层的搜索

(5) 在没出一层递归之后，将 book[curX][curY]还原，将 resultX[length]和 resultY[length]清空，使路径长度减少

(6) 总而言之，深度优先搜索就是从当前的点出发，尝试各个方向的走法，一旦找

到了终点或者走进了死胡同，那么就开始返回，在返回的过程当中重置我们存储的信息。最终我们能够遍历从起点出发的每一条可以行走的道路。

```
void Map::dfs(int x, int y) {
    int curX, curY;
    if (x == destX && y == destY) {
        for (int i = 0; i < length; i++) {
            std::cout << "<" << resultX[i] << "," << resultY[i] << "> ---> ";
        }
        std::cout << "<" << destX << "," << destY << ">\n";
        flag = true;
        return;
    }
    for (int i = 0; i < 4; i++) {
        curX = x + moveDir[i][0];
        curY = y + moveDir[i][1];
        if (curX<0 || curX>=rLength || curY<0 || curY>=cLength) {
            continue;
        }
        if (book[curX][curY] == UNMARKED) {
            if (map[curX][curY] == '.') {

                book[curX][curY] = MARKED;
                length++;
                resultX[length] = curX;
                resultY[length] = curY;

                dfs(curX, curY);

                book[curX][curY] = UNMARKED;
                resultX[length] = 0;
                resultY[length] = 0;
                length--;
            }
        }
    }
    return;
}
```

四、 测试

1、 第一个地图

该地图只有一条路径，起点为<1.1>，终点为<5,5>

```
Microsoft Visual Studio 调试控制台
本程序内置了三个地图，下面是第一个地图：
迷宫1地图：
        0列     1列     2列     3列     4列     5列     6列
0行     #       #       #       #       #       #       #
1行     #       .       #       #       #       #       #
2行     #       .       #       #       #       #       #
3行     #       .       .       .       #       #       #
4行     #       #       #       .       .       .       #
5行     #       #       #       #       #       .       #
6行     #       #       #       #       #       #       #


迷宫1路径：
<1, 1> ---> <2, 1> ---> <3, 1> ---> <3, 2> ---> <3, 3> ---> <4, 3> ---> <4, 4> ---> <4, 5> ---> <5, 5>
```

2、第二个地图

该地图的路径更多了，有四条路径，起点为<1.1>，终点为<5,5>

```
迷宫2地图:
          0列       1列       2列       3列       4列       5列       6列
0行       #         #         #         #         #         #         #
1行       #         .         .         .         .         .         #
2行       #         .         #         #         #         .         #
3行       #         .         .         .         #         .         #
4行       #         #         #         .         .         .         #
5行       #         #         #         #         .         .         #
6行       #         #         #         #         #         #         #


迷宫2路径:
<1,1> ---> <2,1> ---> <3,1> ---> <3,2> ---> <3,3> ---> <4,3> ---> <4,4> ---> <5,4>
---> <5,5>
<1,1> ---> <2,1> ---> <3,1> ---> <3,2> ---> <3,3> ---> <4,3> ---> <4,4> ---> <4,5>
---> <5,5>
<1,1> ---> <1,2> ---> <1,3> ---> <1,4> ---> <1,5> ---> <2,5> ---> <3,5> ---> <4,5>
---> <5,5>
<1,1> ---> <1,2> ---> <1,3> ---> <1,4> ---> <1,5> ---> <2,5> ---> <3,5> ---> <4,5>
---> <4,4> ---> <5,4> ---> <5,5>
```

3、 第三个地图

有两条路径，起点为<1.1>，终点为<3,1>

```
迷宫3地图:
          0列       1列       2列       3列       4列       5列       6列
0行       #         #         #         #         #         #         #
1行       #         .         .         .         .         .         #
2行       #         #         #         #         #         .         #
3行       #         .         .         .         #         .         #
4行       #         #         #         .         .         .         #
5行       #         #         #         #         .         .         #
6行       #         #         #         #         #         #         #


迷宫3路径:
<1,1> ---> <1,2> ---> <1,3> ---> <1,4> ---> <1,5> ---> <2,5> ---> <3,5> ---> <4,5>
---> <5,5> ---> <5,4> ---> <4,4> ---> <4,3> ---> <3,3> ---> <3,2> ---> <3,1>
<1,1> ---> <1,2> ---> <1,3> ---> <1,4> ---> <1,5> ---> <2,5> ---> <3,5> ---> <4,5>
---> <4,4> ---> <4,3> ---> <3,3> ---> <3,2> ---> <3,1>
请按任意键继续. . .
```

3、 第四个地图

起点为<1.1>，终点为<3,1>，如图所示没有这样的通路

```
迷宫4地图:
          0列       1列       2列       3列       4列       5列       6列
0行       #         #         #         #         #         #         #
1行       #         .         .         .         .         .         #
2行       #         #         #         #         #         .         #
3行       #         .         #         .         #         .         #
4行       #         #         #         .         .         .         #
5行       #         #         #         #         .         .         #
6行       #         #         #         #         #         #         #


迷宫4路径:
没有这样的路径
```