

上机报告-4

数算B 谢胡睿 2400014151

题目

1.题目背景

在一个二维迷宫中分布着一些食物，利用数字大小标定食物的多少，数字越大表示食物越多。小明在迷宫中每行进一格需要消耗一份食物。现在需要设计一个算法来帮助小明在迷宫中找到一条路径，使得他能够在某一刻获得理论上最多的食物。小明可以自由选择路径的起始点、终点、途径点等，路径允许交叉。小明可以在某一刻持有负数个食物（可理解为身上有其它的食物，但不计入本题的计算）。 本题建议使用广度优先搜索算法与深度优先搜索算法综合求解

2.题目描述

对于给定的迷宫，求小明能获得的最多食物。

3.输入格式

第一行有两个正整数m，n。其中m代表迷宫的行数，n代表迷宫的列数。接下来m行，每行包含以一个空格分隔的n个数字，表示迷宫的布局。数字“-1”表示墙壁，数字“0”表示可以通行的空地，非零的正数代表该地点存在食物，且越大的数字代表该地点存在越多的食物。迷宫最外侧必定由墙壁围绕，小明只能上下左右进行移动。

4.输出格式

输出一行，一个整数，为理论能够持有的最大食物。

5.样例

输入输出样例

样例1:

input:

```
7 6
-1 -1 -1 -1 -1 -1
-1 5 0 -1 8 -1
-1 0 1 -1 0 -1
-1 0 0 0 0 -1
-1 0 7 -1 0 -1
-1 6 -1 -1 -1 -1
-1 -1 -1 -1 -1 -1
```

output: 15

样例2:

input:

```
6 8
-1 -1 -1 -1 -1 -1 -1 -1
-1 0 -1 0 1 0 10 -1
-1 0 7 0 -1 0 0 -1
-1 0 -1 0 5 -1 0 -1
-1 8 0 0 -1 0 2 -1
-1 -1 -1 -1 -1 -1 -1 -1
```

output: 19

Solution

总体描述

针对这个迷宫食物收集问题，我采用了深度优先搜索(DFS)加回溯的策略。核心思想是从每个可能的起点出发，尝试所有可能的移动路径，并在搜索过程中维护当前获得的最大食物值。

方案一：直接DFS回溯

设计思路

这种方法采用深度优先搜索加回溯的策略。核心思想是从每个可能的起点出发，尝试所有可能的移动路径，并在搜索过程中维护当前获得的最大食物值。

解决方案中使用了以下关键组件：

DFS回溯：探索所有可能路径并回溯 剪枝策略：通过潜力估计（当前值+剩余食物总量）提前终止无希望的搜索路径 状态恢复：回溯时恢复迷宫状态，确保不同路径之间互不干扰 优先级搜索：从食物价值最高的点开始搜索，提高搜索效率

优缺点

优点： 实现简单直观 空间复杂度低 适合中小规模迷宫

缺点： 时间复杂度高，对于大规模迷宫可能超时 搜索空间随迷宫大小指数增长

方案二： BFS + 状态压缩DP

设计思路

这种方法将问题分解为两个阶段：

使用BFS预计算所有食物点对之间的最短距离，构建距离矩阵 使用状态压缩动态规划找出访问所有食物点的最优路径 具体实现步骤：

使用BFS计算每对食物点之间的最短距离 构建状态dp[pos][mask]表示当前在pos位置且已访问mask状态下的最大收益 通过记忆化搜索优化状态转移计算 从每个食物点出发，计算最大可能收益

优点：

时间复杂度低，对于食物点较少的情况效率高 能够处理更复杂的迷宫 可以保证找到全局最优解
最终，我选择了BFS+状态压缩DP方法作为主要解决方案，因为它能处理更复杂的测试用例。

问题与挑战

- 1. 搜索效率问题 问题：对所有可能的起点和路径进行无差别搜索，导致搜索空间过大。

解决方案：

DFS方法：通过优先考虑高价值起点，并实现有效剪枝 BFS+DP方法：通过预计算距离矩阵，将问题转化为图上的组合优化问题 2. 状态表示与转移 问题：如何高效表示和转移已访问的食物点状态。

解决方案：使用位运算进行状态表示和操作，大幅降低空间复杂度和运算成本。

总结 通过实现迷宫食物收集问题的两种解决方案，我深入理解了搜索算法、动态规划和状态压缩技术在路径规划中的应用。