

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。

我设计的解决方案中使用了以下关键组件：

1. DFS回溯：探索所有可能路径并回溯
2. 剪枝策略：通过潜力估计提前终止无希望的搜索路径
3. 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

设计与实现

见代码

问题与挑战

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，效率较低。

解决方案：通过优先考虑有食物的位置作为起点，并按食物价值大小排序，优先尝试高价值起点。

深度限制缺失

问题：对于复杂迷宫，搜索深度可能过大，导致堆栈溢出。

解决方案：添加深度限制参数，控制递归深度。

总结

通过实现迷宫食物收集问题，我深入理解了深度优先搜索、回溯法和剪枝优化在路径规划中的应用。关键收获包括：

在递归搜索中，正确的状态维护和回溯至关重要，尤其是对共享资源的修改。有效的剪枝策略能显著提高搜索效率，减少不必要的计算。