**高级算法设计与分析 课程**

**实验报告**

**实验二：搜索算法**

姓名：陈峰

学号：7203610323

班级：？？？

评分表：（由老师填写）

|  |  |
| --- | --- |
| 最终得分： | |
| 对实验题目的理解是否透彻： | |
| 实验步骤是否完整、可信 ： | |
| 代码质量 ： | |
| 实验报告是否规范 ： | |
| 趣味性、难度加分 ： | |
| 特 色： | 1 |
| 2 |
| 3 |

**一、实验题目概述**

寻路问题。输入一个方格表示的地图，要求用A\*算法找到并 输出从起点（在方格中标示字母S）到终点（在方格中标示字母T）的代价最小 的路径。有如下条件及要求： （1）每一步都落在方格中，而不是横竖线的交叉点。

（2）灰色格子表示障碍，无法通行。

（3）在每个格子处，若无障碍，下一步可以达到八个相邻的格子，并且只可以 到达无障碍的相邻格子。其中，向上、下、左、右四个方向移动的代价为1，向 四个斜角方向移动的代价为√2。

（4）在一些特殊格子上行走要花费额外的地形代价。比如，黄色格子代表沙漠， 经过它的代价为4；蓝色格子代表溪流，经过它的代价为2；白色格子为普通地 形，经过它的代价为0。

（5）经过一条路径总的代价为移动代价+地形代价。其中移动代价是路径上所做 的所有移动的代价的总和；地形代价为路径上除起点外所有格子的地形代价的总 和。比如，在下图的示例中，路径A→B→C的代价为√2+1(移动)+0(地形)，而 路径D→E→F的代价为2(移动)+6(地形)。

**二、对实验步骤的详细阐述**

设计A\*算法的核心就是设计启发式函数，启发式函数要求必须满足以下要求：

1. 非负性。H(n)>0对于任意节点n满足。
2. 一致性。对于任意两个节点n,m,设n到m的实际代价为cost(n,m),启发函数要满足,当满足了一致性，自然也就满足了可采纳性，即 ,启发函数的代码要小于实际代价。

设计双向和单向的A\*算法本质是一致的。

设计的启发函数如下：

按照以下步骤进行A\*算法的流程。

1.  初始化：  
   创建两个集合：开放集合（Open Set）和关闭集合（Closed Set）。  
   将起始节点加入开放集合，并为其分配以下评分：  
     - g(n)：从起始节点到当前节点的实际代价。  
     - h(n)：当前节点到目标节点的估计代价（启发式）。  
     - f(n)：节点的总评分，即 `f(n) = g(n) + h(n)`。  
  
2. 循环以下步骤：  
   - 在开放集合中找到具有最低`f(n)`值的节点，将其称为当前节点。  
   - 将当前节点从开放集合移动到关闭集合。  
   - 检查当前节点是否是目标节点。如果是，则重构路径并结束搜索。  
   - 对于当前节点的每个邻居节点：  
     - 如果邻居节点在关闭集合中，跳过不处理。  
     - 计算邻居节点的`g(n)`值。如果这个邻居节点已经在开放集合中，并且通过当前节点到达它的新路径不比原有的路径更优，也跳过不处理。  
     - 如果邻居节点不在开放集合中，或者新的路径更优，更新邻居节点的`g(n)`，`h(n)`，和`f(n)`值，并将其父节点设置为当前节点。  
     - 如果邻居节点不在开放集合中，将其加入开放集合。  
  
3. 结束条件：  
   - 如果开放集合为空且没有找到目标节点，则路径不存在。  
   - 如果找到目标节点，按照父节点指针从目标节点回溯到起始节点，这条路径即为最优路径。

对于双向的A\*算法同理，区别在于需要设定两个openlist、closelist用于区别前向和后向。

**三、实验数据**

**1. 实验设置**

**实验环境**：

Windows 11，vscode、python

**数据**：

给定的两个测试地图。

**2. 实验结果**

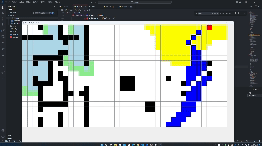


图 1 单向

绿色表示openset内的点，浅蓝色表示closeset内的点，最终显示的是路径。

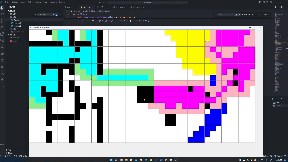


图 2 双向

前向和后向分别采用了不同的颜色来进行区分。

上述的运行结果为了展现，在运行过程中使用了sleep(0.02)的操作。

**四、对实验结果的理解和分析**

1. A\*算法中的启发式函数的设计是整个算法的核心，启发函数需要尽可能接近实际的代码，这样整个算法的运行才能具有高效率**。**

2. 在进行A\*算法的时候，也可以采用一些其他的搜索所具备的剪枝策略，例如可行性、最优性剪枝策略等。

上述单向和双向得到的路径不同，但是路径的代价都是一致的，所以是得到最优解，但是最优解并不止一个。

**五、实验过程中最值得说起的几个方面**

1. 使用了python实现了单向和双向的A\*算法，并且使用了Python的Tkinter来进行可视化，加强了对于A\*算法的理解，使得对于算法的理解更加形象和直观。