

人工智能基础

编程作业 1

<http://staff.ustc.edu.cn/~linlixu/ai2019spring/ai2019spring.html>

完成截止时间：2019/5/12

提交至：ustc_ai2019@163.com

助教： 盛鑫 [xins@mail.ustc.edu.cn]

赵若宇 [zry1997@mail.ustc.edu.cn]

滕思洁 [yunmo@mail.ustc.edu.cn]

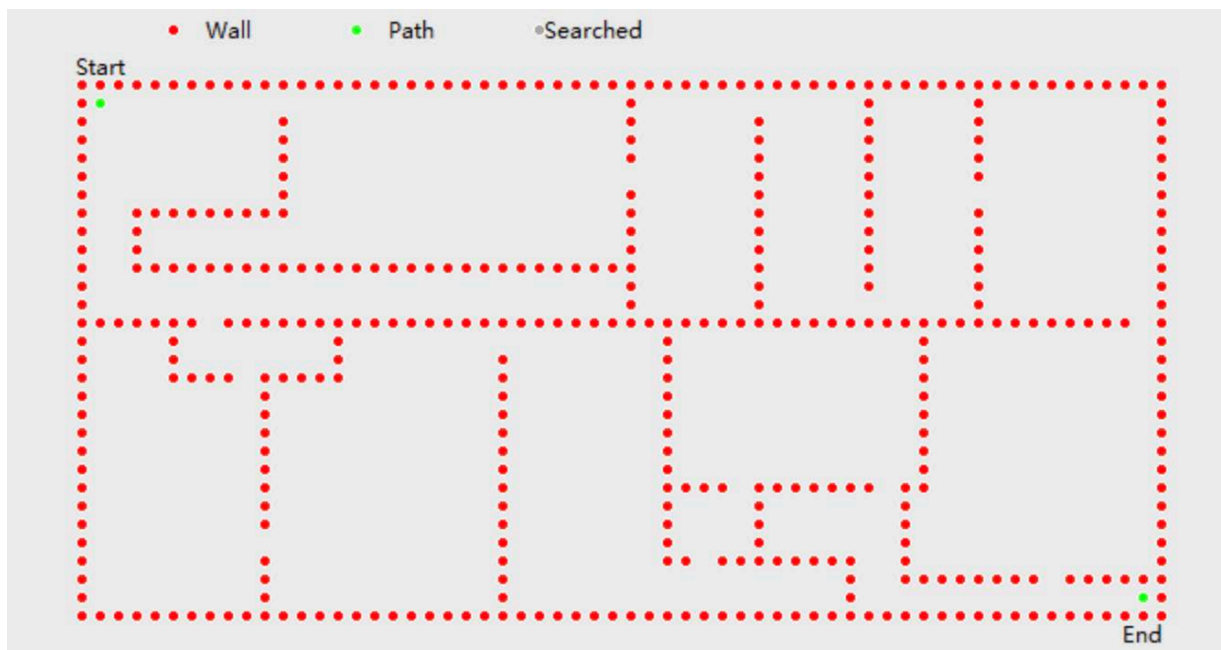
辛媛 [lwjxy@mail.ustc.edu.cn]

A*搜索问题 (10point)

问题描述：

迷宫问题可以表述为：下图是一个迷宫，**红色圆点**代表墙，**绿色原点**代表寻路者。图中 start 和 end 分别代表初始和目标位置。空格代表可以通行的路径。（本次实验所需的迷宫用矩阵表示，详见作业要求第一条）。

要求找出从起始单元格出发，到达目标单元格的、所走过的单元格序列。注意寻路者只能移动到与其行相邻或列相邻的单元格。在任一个单元格中，都只能看到与它邻近的 4 个单元格(如果位于底边，则只有 3 个；位于 4 个角上，则只有 2 个)是否能通过。



实验方法:

A*算法实现

A*算法是人工智能中的一种搜索算法，是一种启发式搜索算法，它不需遍历所有节点，只是利用包含问题启发式信息的评价函数对节点进行排序，使搜索方向朝着最有可能找到目标并产生最优解的方向。它的独特之处是检查最短路径中每个可能的节点时引入了全局信息，对当前节点距终点的距离做出估计，并作为评价节点处于最短路线上的可能性的度量。

IDA*算法

IDA*算法即迭代加深的 A*算法。IDA*的基本思路是：首先将初始状态结点的 f 值设为阈值 maxf，然后进行深度优先搜索，搜索过程中忽略所有 f 值大于 maxf 的结点；如果没有找到解，则加大阈值 maxf，再重复上述搜索，直到找到一个解。在保证 f 值的计算满足 A*算法的要求下，可以证明找到的这个解一定是最优解。在程序实现上，IDA* 要比 A* 方便，因为不需要保存所有结点，不需要判断重复，也不需要根据 f 值对结点排序，占用空间小。而这里在 IDA*算法中也使用合适的评价函数，来评估与目标状态的距离。

作业要求:

1.迷宫大小设置为 18*25，所以输入为大小为 18*25 二维数组 A[18][25]，数组的下标代表点的位置，数组的值代表该位置是否可通行，0 表示可通行，1 表示不可通行。本实验中入口和出口的位置设为(1,0) 和(16,24)；需要的迷宫矩阵 input.txt 已经给出。

[illegible]

0.0000334
RRDDRD
6

输入文件 (input.txt)

输出文件 (output.txt)

2. 输出时将花费的时间（以 s 为单位），动作序列，总步数输出到文件。字母大写，字母之间无空格。U 代表 up，即上移，D 代表 down，即下移，L 代表 left，即左移，R 代表 right，即右移。输出的动作序列应为从初始状态(入口)开始，到目标状态（出口）结束时，中间经过的所有空格的操作动作。在测试时，我们会从输入状态开始，执行你的输出动作序列，看你的动作序列是否合法，通过此动作序列能否到达出口等。

3. 挑选你认为**性能最优**的启发函数，比如曼哈顿距离或欧式距离，但不限于这两个，在报告中说明你的选择并给出理由。使用 C/C++ 实现 A* 和 IDA* 的 2 个算法，对应输出文件为 **output_A.txt, output_IDA.txt**。

4. **交源代码和可执行文件**(2 个算法所以有 2 个代码和可执行文件)，若使用命令行编译请给出编译命令。在实验报告(word 或 pdf)中写明如何运行 你的程序以及对每个程序的说明。并大致说明你算法(A*和迭代 A*)的**时间复杂度和空间复杂度**。使用表格列出我们所供的样例的结果，包括步数和运行时间(如果对于某些样例，你的代码无法得出结果，标记为×)，我们对时间或者空间优化相应加分。

5. **严禁抄袭**，我们会用软件进行代码查重，2 个算法都要求实现，我们会查看源代码，严禁只实现一个算法。一旦发现上述情况，以 0 分计。

五子棋人机对弈 (10point)

五子棋是世界智力运动会竞技项目之一，是一种两人对弈的纯策略型棋类游戏，是世界智力运动会竞技项目之一，通常双方分别使用黑白两色的棋子，下在棋盘直线与横线的交叉点上，先形成 5 子连线者获胜。我们这里使用最简单的五子棋规则如下：

- 1、黑白双方一人走一次。
- 2、要一方的五颗子能够在棋盘内连成一条线，则该方获胜。

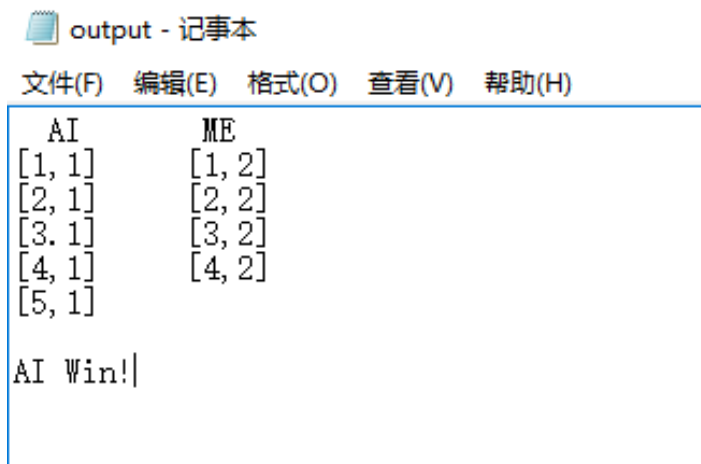
围棋棋盘使用国际比赛标准，为十五路（15×15）棋盘，形状近于正方形，平面上画横竖各 15 条平行线，线路为黑色，构成 225 个交叉点。

实验任务：

1. 设计一个**评分函数**对棋盘上局面进行评分。请在实验报告中展示你的评分规则并给出理由。
2. 利用你设计好的评分函数生成一颗**博弈树**。使用 **minimax 算法和 Alpha- Beta 剪枝策略**实现一个固定搜索深度（搜索深度大于 1）的人机对弈的五子棋 AI。
3. 结果的呈现为你和你的 AI 棋手对弈一局的过程。
4. 思考题：
 - 1) 思考搜索的深度对 AI 的决策效率有何影响？如何利用搜索深度提高 AI 的智能程度？
 - 2) Alpha-Beta 剪枝法在减枝过程搜索效率与节点的排列顺序有很大关系。思考是否可以改进剪枝策略提高决策速度？
 - 3) 思考是否有方法实现 AI 的自学习能力，让 AI 不在相同的地方犯错？本题只需要给出思路，不需要具体实现。

作业要求：

1. 实验输出要求输出到文本，文件名为 output.txt，输出格式为两列，第一列为 AI 棋手落子坐标（默认 AI 先手下棋），第二列为你落子的坐标，在一方获胜时给出胜负判断标志。



```
output - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
AI      ME
[1, 1]  [1, 2]
[2, 1]  [2, 2]
[3, 1]  [3, 2]
[4, 1]  [4, 2]
[5, 1]  [4, 2]
AI Win!|
```

2. 使用 C/C++ 实现上述算法。提交时请提交**源代码和可执行文件**，若使用命令行编译请给出编译命令。
3. 实验报告使用 PDF 格式提交。实验报告中包含以下几点：
 - 1) 你设计的评分函数和理由。
 - 2) 算法思想，分析 Alpha-Beta 剪枝法在本实验中的作用。
 - 3) 实验结果说明，分析你的 AI 棋手的棋力大小，在和 AI 对弈的过程中你的棋力是否提高
 - 4) 要求回答思考题中的任意两题，三题都回答按照回答质量适当加分。

实验提交

在截止时间之前将作业提交到 ustc_ai2019@163.com,邮件主题为“**学号_姓名_实验一**”。提交后会收到确认接受的邮件，以此邮件为准。

将“**A*搜索问题**”和“**五子棋 AI**”所需要提交的文件分别放在“**A*搜索问题**”和“**五子棋 AI**”文件夹中，这两个文件夹放在“**学号_姓名_实验一**”文件夹中，压缩成“**学号_姓名_实验一.zip**”。将压缩包作为邮件附件一起提交到指定邮箱。

务必按时提交实验，不接受逾期提交的实验。

实验中有任何问题请直接联系助教