

人工智能基础

实验 1

完成截止时间：2019/4/30/23:59

提交至：ailab_2019@163.com

助教：姜庆彩【QQ：649561941】

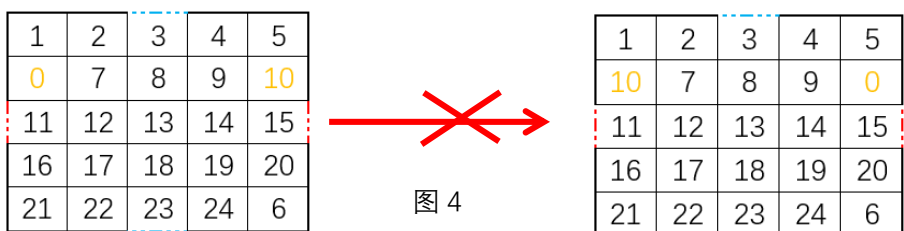
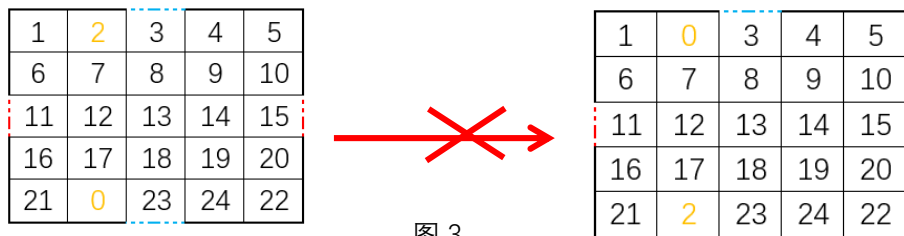
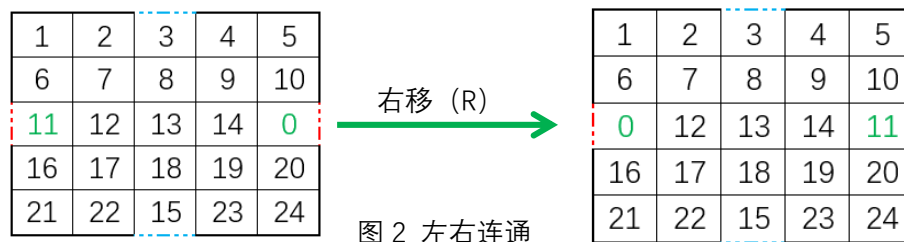
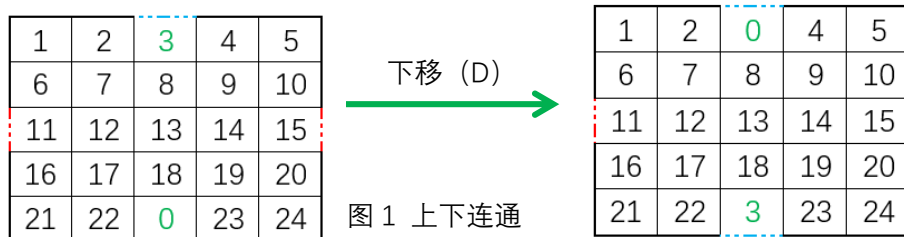
林思捷【QQ：419803495】

梁聪【QQ：386166518】

Problem1: 24 数码问题 (50%)

本问题包括一个 5×5 的表格, 24 个写有数字(1-24)的棋子以及一个空位(由 0 表示), 两个棋盘边缘的连通点 (边框红色和蓝色的位置)。与空位上、下、左、右相邻的棋子可以移动到空位中。

连通点为：蓝色连通点可上下互达，红色连通点可左右互达，如下图中，图 1 与图 2 为合法移动，图 3 与图 4 为不合法移动。游戏的目的是要达到一个特定的目标状态。



问题表示：

本次作业中，初始状态和目标状态都由一个 2 维矩阵表示，0 表示空位置，1-24 表示棋子，本次作业红蓝连通点位置固定，初始状态和目标状态都通过文件输入。如图 5 图 6 所示，为一个例子。

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 4 | 9 | 5 |
| 6 | 7 | 8 | 0 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 3 | 23 | 24 |

图 5 初始状态举例

| | | | | |
|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 0 |

图 6 目标状态举例

定义空格棋子 0 的 4 个动作，U 代表 up，即对棋子上移，D 代表 down，即对棋子下移，L 代表 left，即对棋子左移，R 代表 right，即对棋子右移。**所有动作均要合法。**

本作业中，需要读取初始状态及目标状态，并实现两个求解八数码问题的算法：**A***搜索及迭代深入**A***搜索(**IDA***)，使用以下两种启发函数：

- $h1(n) = \text{number of misplaced tiles}$ (错位的棋子数)
- 自己设计一种比 $h1$ 更好的可采纳启发式，要求在实验报告中证明其可采纳性，**注意曼哈顿距离不可采纳。**(提示：可以考虑错误棋子数、曼哈顿距离、欧式距离等的变种)

最后输出从初始状态到目标状态的动作序列，例如图 5 的初始状态到达图 6 的目标状态的移动序列是 **ULURR**。

迭代**A***搜索算法的提出是为了解决**A***搜索在空间复杂度上的缺点，将迭代深入的思想用在启发式搜索上。**IDA***和典型的迭代深入算法最主要的区别就是所用的截断值是 f 耗散值 ($g+h$) 而不是搜索深度；每次迭代，截断值是超过上一次迭代阶段值的节点中最小的 f 耗散值。以下为迭代**A***搜索算法。

Algorithm 3 Iterative deepening A* search (IDA*)

```
1:  $\hat{d\_limit} \leftarrow \hat{d}(s_0)$ 
2: while  $\hat{d\_limit} < \infty$  do
3:    $next\_d\_limit \leftarrow \infty$ 
4:    $list \leftarrow \{s_0\}$ 
5:   while list is not empty do
6:      $s \leftarrow head(list)$ 
7:      $list \leftarrow rest(list)$ 
8:     if  $\hat{d}(s) > \hat{d\_limit}$  then
9:        $next\_d\_limit \leftarrow \min(next\_d\_limit, \hat{d}(s))$ 
10:    else
11:      if  $s$  is a goal then
12:        return  $s$ 
13:      end if
14:       $newstates \leftarrow \text{apply actions to } s$ 
15:       $list \leftarrow \text{prepend}(newstates, list)$ 
16:    end if
17:  end while
18:   $\hat{d\_limit} \leftarrow next\_d\_limit$ 
19: end while
20: return fail
```

图 7 迭代深入 A*算法

实验要求：

1. 统一从文件输入，文件名为 input.txt,目标状态文件名为 target.txt, 注意检查初始状态与目标状态是否匹配，每个算法输出到文件对应文件 Ah1_solution.txt (对应A*搜索算法使用 h1 启发式, 下面以此类推)、IDAh1_solution.txt、Ah2_solution.txt、IDAh2_solution.txt, 所有文件和对应算法的可执行文件在同一个文件夹。
2. 输入格式为 5×5 矩阵格式，输出第一行为算法运行时间（单位秒/s），第二行为移动序列，第三行为总步数。例如，对于图 5 的示例初始状态文件和图 6 的目标状态文件内容如下：

| input.txt - 记事本 | target.txt - 记事本 | solution.txt - 记事本 |
|---------------------|---------------------|--------------------|
| 文件(F) 编辑(E) 格式(O) 查 | 文件(F) 编辑(E) 格式(O) 查 | 文件(F) 编辑(E) 格式(O) |
| 1 2 4 9 5 | 1 2 3 4 5 | 0.00063s |
| 6 7 8 0 10 | 6 7 8 9 10 | ULURR |
| 11 12 13 14 15 | 11 12 13 14 15 | 5 |
| 16 17 18 19 20 | 16 17 18 19 20 | |
| 21 22 3 23 24 | 21 22 23 24 0 | |

图 8 输入输出文件举例

输入文件每行数字之间由若干空格隔开，行末回车，输出文件用单行存储移动序列。请严格按照该格式进行输入以便助教批改。

- 实验报告中请使用我们提供的测试用例进行分析，见 [24 数码问题状态.docx](#)，列表登记每个样例的运行结果，包括样例编号、运行时间、移动序列、总步数，如果对于某个样例你的代码无法得到结果，登记为 (×)。测试用例旁注释的参考步数不一定是最佳步数。批改时我们会从初始状态开始检测每一步移动是否合法。
- 编程语言限制为 c、c++、python、java 其中任选，源码中尽量注释清楚。实验报告需要对代码关键部分进行解释，并写明如何运行你的程序，如果需要命令行编译请写明编译命令。
- 实验报告中大致说明你算法的时间复杂度和空间复杂度，对运行空间和时间的优化都会有加分，请在实验报告中说明优化方法。优化有利于让长步数的样例跑出结果。
- 严禁抄袭，批改时会进行代码查重，抄袭以 0 分记。一共要实现 4 个算法，禁止仅实现不足 4 个算法然后冒充其他算法，每缺一个算法此部分将扣 25% 分数。

Problem2: 数独问题 (50%)

原始数独问题：在 9×9 的棋盘上每格填入一个 1~9 的数字，使得填满棋盘后每行和每列都不存在重复数字，每个粗线划分的小九宫格内也不存在重复数字。例如下图所示，左边是初始棋盘，右边是一个可行解。

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 9 | | | | 6 | | |
| | 6 | | 1 | | 4 | | 9 | |
| 4 | | | | 3 | | | | 7 |
| | 4 | | | | | | 8 | |
| | | 5 | | | | 1 | | |
| | 1 | | | | | | 5 | |
| 2 | | | | 8 | | | | 6 |
| | 5 | | 6 | | 3 | | 4 | |
| | | 1 | | | | 3 | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 7 | 9 | 8 | 5 | 2 | 6 | 3 | 4 |
| 5 | 6 | 3 | 1 | 7 | 4 | 2 | 9 | 8 |
| 4 | 2 | 8 | 9 | 3 | 6 | 5 | 1 | 7 |
| 3 | 4 | 6 | 2 | 1 | 5 | 7 | 8 | 9 |
| 7 | 9 | 5 | 3 | 4 | 8 | 1 | 6 | 2 |
| 8 | 1 | 2 | 7 | 6 | 9 | 4 | 5 | 3 |
| 2 | 3 | 4 | 5 | 8 | 1 | 9 | 7 | 6 |
| 9 | 5 | 7 | 6 | 2 | 3 | 8 | 4 | 1 |
| 6 | 8 | 1 | 4 | 9 | 7 | 3 | 2 | 5 |

图 9 数独问题举例

我们修改问题为：在 $n \times n$ 的棋盘上每格填入一个 1~k 的数字，使得填满棋盘后每行和每列都不存在重复数字，如图 10 图 11 所示。

| | | | |
|---|---|---|---|
| 1 | | | |
| | | 5 | |
| | 2 | | |
| | | | 2 |

| | | | |
|---|---|---|---|
| 1 | 5 | 4 | 3 |
| 2 | 4 | 5 | 1 |
| 3 | 2 | 1 | 5 |
| 4 | 1 | 3 | 2 |

图 10 $n=4, k=5$ 问题举例

| | | | | | |
|---|---|---|---|---|---|
| 1 | | | | 6 | |
| | | 3 | | | |
| | 3 | | | | |
| | | | 1 | | |
| | | | | | 3 |
| | | 4 | | | |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 2 | 3 | 6 | 5 |
| 4 | 2 | 3 | 6 | 5 | 1 |
| 2 | 3 | 6 | 5 | 1 | 4 |
| 3 | 6 | 5 | 1 | 4 | 2 |
| 6 | 5 | 1 | 4 | 2 | 3 |
| 5 | 1 | 4 | 2 | 3 | 6 |

图 11 $n=6, k=6$ 问题举例

问题描述：

给定 $n \times n$ 的初始棋盘与限制值 k , $k \geq n$, 要求：

1. 使用不小于 1 不大于 k 的整数填充棋盘；
2. **不得改变初始棋盘已有的数字；**
3. 棋盘填满后每行与每列均不得有重复数字，输出**一个**可行解。

从 3 个算法中选 2 个实现：CSP 问题的前向检验算法、遗传算法、模拟退火算法。也可以从上述 3 个算法中选取一个后，再自己另找一个或设计一个 CSP 问题的局部搜索算法来实现。如果你另外的搜索算法比上述 3 个算法好，可以加分。

实验要求：

1. **统一从文件输入输出**，输入文件 input.txt 第一行第一个数字为 n 值，第二个数字为 k 值，第 2 行到第 $n+1$ 行为棋盘矩阵，同行数字用空格分割，行末回车。输入矩阵中用 0 表示待填充空格。输出为 algorithm_solution.txt，第一行为算法用时，第 2 行至第 $n+1$ 行为结果矩阵。**所有文件均与对应算法源码目录相同。**上方图 10 的示例对应的输入输出文件如下图所示。

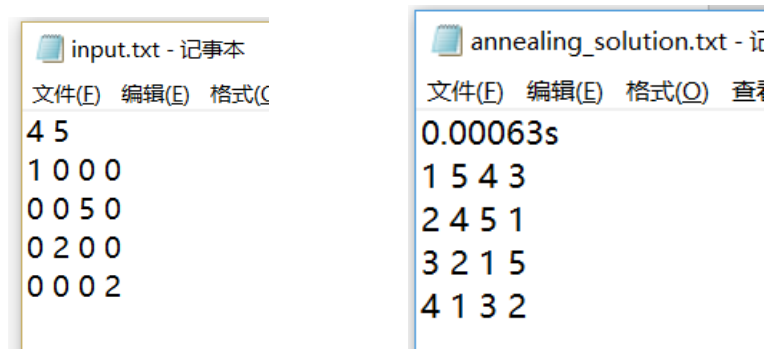


图 12 输入输出样例

2. **编程语言限制为 c、c++、python、java** 其中任选，源码中尽量注释清楚。实验报告需要对代码关键部分进行解释，并写明如何运行你的程序，如果需要命令行编译请写明编译命令。
3. 实验报告中请**记录在我们提供的样例上的测试结果**，见[数独问题输入.docx](#)，包括**样例编号、平均运行时间、每个初始状态记录一个解矩阵**，如果你的算法对于某个样例无法得到合法结果，登记为 (×)。 n 和 k 的规模都不超过 20。
4. 源代码尽量注释清楚，实验报告中要说明：算法思想；关键代码说明；如何提升速度。

速度越快给分相应提高。如果使用了非 3 个给定算法，请对算法进行详细说明，最好有对算法时空复杂度的分析。

5. 严禁抄袭，批改时会进行代码查重，抄袭以 0 分记。一共要实现 2 个算法，禁止仅实现 1 算法然后冒充其他算法，每缺一个算法此部分将扣 50% 分数。

实验提交：

在截止时间之前将压缩包发送到 ailab_2019@163.com，邮件主题为**学号_姓名_lab01**，如果需要重复提交则主题为**学号_姓名_lab01_重交 n**，其中 n 为重交次数。如果有重交则之前提交作废。如果没有收到自动回复请联系助教。

压缩包命名为**学号_姓名_lab01**（.压缩包后缀），仅限包含一个同名文件夹**学号_姓名_lab01**。2 个问题各需要 1 份 pdf 格式实验报告，所有相关文件分别打包在**24 数码**和**数独**文件夹内，2 个文件夹放于**学号_姓名_lab01**文件夹内，建议在**24 数码**和**数独**文件夹内对每个算法建立子文件夹。

逾期提交则本次实验得分 $\times 0.7$ 。5 月 12 日 23:59:59 之后不接受补交。

对实验有任何问题请尽快联系助教