

编程作业1

22数码问题

N皇后问题

P1:22数码问题

(50%)

- 本问题包括一个 5×5 的表格，22个写有数字(1-22)的棋子以及一个空位（由0表示），两个**半透明障碍位**（涂黑的部分，可以用-1表示）。与空位上、下、左、右相邻的棋子可以移动到空位中。
- 半透明障碍位**为：上下不可穿透，左右可穿透，如下图中，图1为合法移动，图2为不合法移动。游戏的目的是要达到一个特定的目标状态。

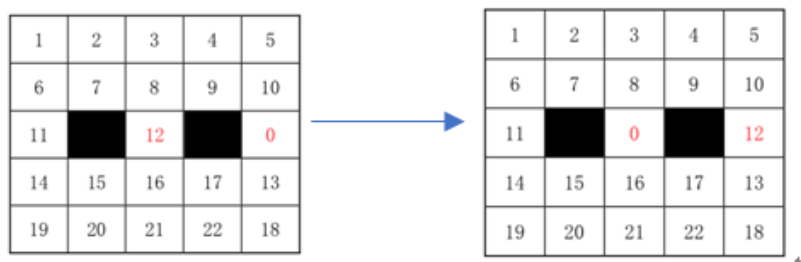


图 1：合法移动(半透明障碍物可左右穿透)

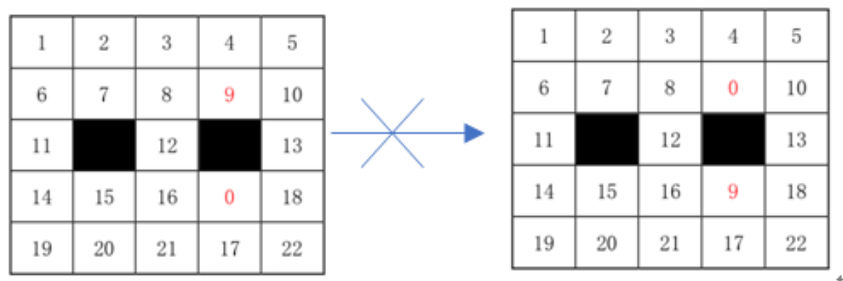


图 2：不合法移动(半透明障碍物不可上下穿透)

22数码问题

1. 实现A*搜索, as
2. 实现迭代深入A*搜索, idas
 - $h1(n)$ = number of misplaced tiles (错位的棋子数)
 - $h2(n)$ = total Manhattan distance (所有棋子到其目标位置的两个方向距离和)

Iterative deepening A*

Problem: A* is space inefficient

IDA*: Iterative deepening on f bound.

Algorithm 3 Iterative deepening A* search (IDA*)

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

问题表示

状态由一个2维矩阵表示，0表示空位置，1-22表示棋子，-1表示半透明障碍物，本次作业半透明障碍物的位置固定(如图所示)，初始状态和目标状态都通过文件输入。如图所示。

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

图 3：22 数码问题的目标状态

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

图 4：22 数码问题的一个初始状态

问题表示

定义4个动作

- U代表up，即对空位0棋子上移
- D代表down，即对空位0棋子下移
- L代表left，即对空位0棋子左移
- R代表right，即对空位0棋子右移
- 所有动作均要合法

问题的目标是找到从初始状态到目标状态需要的**最短**移动序列
输出为动作序列，如 **RDDRR**

作业要求

- 实现4个算法，即，使用启发函数 $h_1(n)$ 的A*算法，使用启发函数 $h_2(n)$ 的A*算法，使用启发函数 $h_1(n)$ 的IDA*算法，使用启发函数 $h_2(n)$ 的IDA*算法。
- 提交源代码和可执行文件（4个算法所以有4个代码和可执行文件），用readme文件写明如何运行你的程序以及对每个程序的说明。
- 大致说明算法（A*和迭代A*）的时间复杂度和空间复杂度。比较使用不同的启发函数 h_1, h_2 的A*搜索及迭代深入A*搜索的性能，并分析性能差异的原因。

P2: N-Queen—N皇后问题 (50%)

本次实验要求： 棋盘中存在M对皇后会互相攻击，同时，其余的皇后之间依然按“N皇后”问题所描述的要求不能互相攻击。

问题描述

- 有一个 $N \times N$ 的棋盘，棋盘上第 i 行（ i 从0开始）第 j 列（ j 从0开始）的位置，记为 (i, j) ，其中 i 、 j 分别为行坐标和列坐标。
- 现有 N 个皇后，请按照以下规则摆放皇后：
 - 1) 存在且仅存在 M 对皇后互相攻击，即这 M 对皇后中每对的行坐标相同，或列坐标相同，或分布在同一斜线上；
 - 2) 对于除1)中的互相攻击关系外，棋盘上不能存在其他的互相攻击的关系。

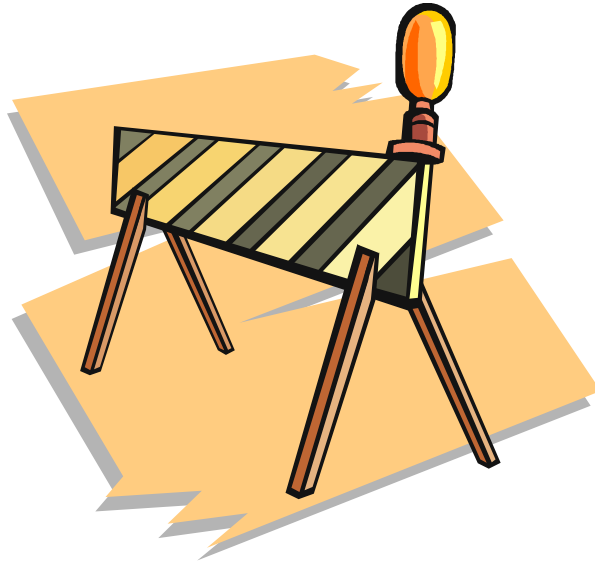
作业要求

从四个算法中选两个算法实现：爬山算法、遗传算法、模拟退火算法以及CSP问题的局部搜索算法。如果可以找到其他更好的基于搜索的算法并实现，可以考虑适当加分。

说明文档要求：

- 算法思想
- 算法如何节省存储空间，分析空间复杂度
- 算法如何提升效率，分析时间复杂度，速度越快给分相应提高，最好能给出**规模-时间**图
- 实验结果说明
- 文档保存为pdf格式

Caution



Academic Integrity