

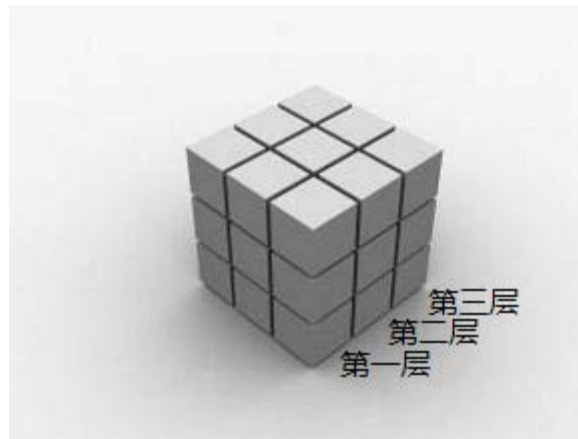
编程作业1

三立方数码问题

2N皇后问题

P1:三立方数码问题 (50%)

三立方数码问题—本问题包括一个 $3 \times 3 \times 3$ 的立方体，包括24个写有数字(1-24)的单位立方体以及一个空位（由0表示），两个障碍位（涂黑的部分，可以由-1表示）。与空位上、下、左、右、前、后相邻的棋子可以滑动到空位中，任何棋子都不能移动到障碍位中，且障碍位不可移动。游戏的目的是要达到一个特定的目标状态。



三立方数码问题

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三个方向距离和)

Iterative deepening A*

Problem: A* is space inefficient

IDA*: Iterative deepening on f bound.

Algorithm 3 Iterative deepening A* search (IDA*)

```
1:  $\hat{d\_limit} \leftarrow \hat{d}(s_0)$ 
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
```

问题表示

状态由一个3维矩阵表示，0表示空位置，1-24表示棋子，-1表示障碍物，如图所示。其余0-24数字任意放置。

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

问题表示

定义6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
- F代表forward，即对空位0棋子前移（靠近自己的方向）
- B代表back，即对空位0棋子后移（远离自己的方向）
- 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 LDBRRB

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: 2N-Queen—2N皇后问题 (50%)

本次实验要求： N个黑皇后和N个白皇后同时存在于棋盘上，相同颜色的棋子不能相互攻击。

问题描述

- 有一个 $N \times N$ 的棋盘，棋盘中第 i 行（ i 从1开始）第 j 列（ j 从1开始）的位置，记为 (i, j) ， i, j 分别为行坐标和列坐标。
- 棋盘上摆放 N 个黑皇后和 N 个白皇后，任意两个黑皇后都不能处于同一行、同一列或同一斜线上，任意两个白皇后都不能处于同一行、同一列或同一斜线上。

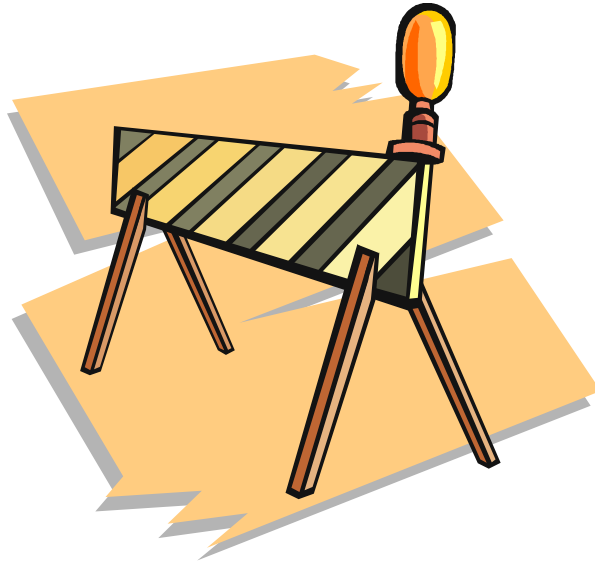
作业要求

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
- 算法效率，分析时间复杂度，速度越快给分相应提高
- 实验结果说明
- 文档保存为pdf格式

Caution



Academic Integrity