

# 编程作业1

十二数码问题

N皇后问题

# P1: 12-puzzle (40%)

**Augmented 12-puzzle** — 本问题包括一个5x3的棋盘，12个写有数字的棋子以及一个空位（由0表示），两个障碍位（由-1表示）。与空位上、下、左、右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。

2	0	3
1	-1	5
4	6	7
8	-1	9
10	11	12

**Initial State**



0	2	3
1	-1	5
4	6	7
8	-1	9
10	11	12



1	2	3
0	-1	5
4	6	7
8	-1	9
10	11	12



1	2	3
4	-1	5
0	6	7
8	-1	9
10	11	12

**Goal State**

# 12-puzzle

1. 实现A\*搜索, as(15%)
2. 实现迭代深入A\*搜索, idas (20%)
  - $h1(n)$  = 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水平竖直距离和)

# Iterative deepening A\*

Problem: A\* is space inefficient

IDA\*: Iterative deepening on d 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
```

---

# 问题表达

输入格式为两个5\*3的矩阵格式

```
2 0 3
1 -1 5
4 6 7
8 -1 9
10 11 12

1 2 3
4 -1 5
0 6 7
8 -1 9
10 11 12
```

# 问题表达

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

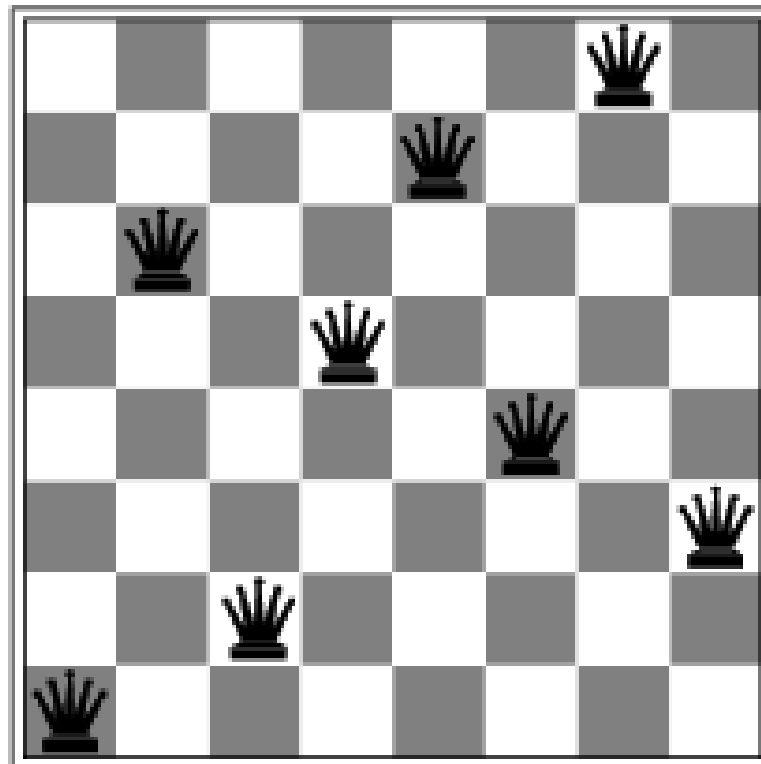
问题的目标是找到从初始状态到目标状态需要的**最短**移动序列  
输出为动作序列，如 **L D D R R U U U L L D D R R D**

# 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。
- 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 (60%)

$N \times N$ 的棋盘上摆放 $N$ 个皇后，使其不能相互攻击，任意两个皇后都不能处于同一行、同一列或同一斜线上。





# 问题表示

在向量 $try[1...N]$ 中存放第1行至第N行皇后摆放的列坐标。即 $try[i]$ 表示第i个皇后摆放在 $(i, try[i])$ 位置上。

若N个皇后不相互攻击，则称 $try[1...N]$ 是N-promising的。

形式化：

$$\forall i, j \in [1, N], \text{ if } i \neq j, \text{ then } try[i] - try[j] \notin \{i - j, 0, j - i\}$$

- 无行冲突，因为第i个皇后放在第i行
- 无列冲突， $try[i]-try[j]$ 不为0
- 无135° 对角线冲突：若第i行和第j列的皇后有此冲突，则 $try[i]-try[j] = j-i$ 。
- 无45° 对角线冲突：若第i行和第j列的皇后有此冲突，则 $try[i]-try[j] = i-j$ 。

# 作业要求

## Part1.实现一个N皇后问题的算法（25%）

```
size_t *NQueen(size_t Queen_number)
```

输入：皇后个数N。

输出：一个满足问题要求的皇后摆放位置序列try[1...n]

## Part2.空间复杂度：可解决的问题规模(N)越大越好（5%）

## Part3.时间复杂度：在给定规模(N)下，解决速度越快越好（30%）

- 采用随机算法的同学，我们将测试一定次数对结果取平均值作为最终结果。
- 请将你的算法如何节省存储空间，提升算法效率的思想写在说明文档内。
- **提交格式：**将源代码，说明文档，实验报告，**release**版的运行程序(.exe)编译好压缩成rar/zip/7z提交。压缩文件文件名为学号+姓名，运行程序文件名为学号（字母大写）。
- No parallel

# Caution



## Academic Integrity