

中山大学数据科学与计算机学院
计算机科学与技术专业-人工智能
本科生实验报告

(2018-2019 学年秋季学期)

学 号： 16337113
姓 名： 劳马东
教学班级： 教务 2 班
专 业： 超算

一、实验题目

- 1、使用 A* 与 IDA* 算法解决 15-Puzzle 问题，启发式函数可以自己选取，最好多尝试不同的启发式函数；
- 2、代码要求使用 python 或者 C++；
- 3、报告要求
 - (a) 报告中需要包含对两个算法的原理解释
 - (b) 需要包含性能和结果的对比和分析
 - (c) 如果使用了多种启发式函数，最好进行对比和分析
 - (d) 需要在报告中分情况分析估价值和真实值的差距会造成算法性能差距的原因

二、实验内容

(一) 算法原理

1、拼图问题的形式化

(1) 问题定义

- 初始状态
- 行动
- 状态空间
- 目标测试
- 路径耗散

(2) 问题的解：序列

(二) 关键代码

```
// 用于表示数码的位数，0-15 一个数字只需4位
#define BITWISE 4
// 1111，用于取低4位
#define LOWERBIT 15
class Puzzle {
    // 16个数刚好用去64位
    __int64 _puzzle;
public:
    const int dim_size;

    Puzzle(int numbers) : dim_size(sqrt(numbers + 1) + 0.5), _puzzle(0) { }

    // 实现类似于二维数组下标索引的get、set接口
    int get(int i, int j) const {
        // 计算偏移量，与位数相乘，得到(i,j)元素最低位的地址
        i = (i * dim_size + j) * BITWISE;
        // 右移，取低4位就是(i,j)元素
```

```
        return (_puzzle >> i) & LOWERBIT;
    }

    void set(int i, int j, int x) {
        __int64 x64 = LOWERBIT;;
        int offset = (i * dim_size + j) * BITWISE;
        // 先将 (i,j) 元素置为0
        _puzzle &= ~(x64 << offset);
        x64 = x;
        // 左移取或就能把 (i,j) 置为x
        _puzzle |= (x64 << offset);
    }
}
```

表 1. 运行时间 (/秒)

搜索策略	样例 1	样例 2	样例 3	样例 4
A*	550	14	808	39

表 2. 生成节点数

搜索策略	样例 1	样例 2	样例 3	样例 4
A*	81004390	2662191	117201754	6927246

三、实验结果及分析

(一) 实验结果展示

(二) 评测指标展示及分析

四、思考题