

算法设计与分析实验报告

实验名称： 骑士巡游问题

一、问题陈述，相关背景、应用及研究现状的综述分析

1.问题陈述：

在一张国际象棋棋盘上（8*8 方格），骑士（knight，马）位于任意一个位置。问如何才能让骑士不重不漏的经过棋盘上的每个格？本问题中已知骑士位置(m,n)，其中 $0 \leq m,n \leq 8$ ，要求给出骑士行走路径，路径可用 8*8 矩阵输出，其中值表示骑士到达此位置行走的步数（初始为 1）；

2.相关背景：

骑士巡游问题也属于 TSP（TravelingSalesmanProblem），最早由数学家欧拉于 1759 年所提出。

3.应用及研究现状：

骑士巡游问题应用较广，如基于骑士巡游的灰度图像加密压缩算法，基于骑士巡游的 Mesh 光网络链路故障定位策略等。

常见的骑士巡游问题的解法有回溯递归

二、模型拟制、算法设计和正确性证明

1. 模型拟制

用二维数组来代表棋盘，初始化全为 0，其中存储的数字为路径的序号，如果为 0 则表示还没有走过。

2. 算法设计

1. 递归解法：从起始点开始，依次探测 8 个方向的点是否可行，如果可行，则在二维数组中存下路径序号，如果不可行，则进行回溯，直到走完所需路径，表明一次巡游完成。

2. 贪心解法：从起点开始，探测周围 8 个方向的点的对应可行路径数，选择其中路径数最少的那一个点，存储路径序号，然后重复上述步骤，继续探索，直到结束。

3. 正确性证明：

递归解法类似于穷举，探索所有路径，直到有一种可行方案出现，但这样的话时间复杂度较高。

贪心解法某些情况下可以求得最优解，但是有些情况只能求得较优解，不过时间复杂度较低。

三、时间和空间复杂性分析

算法 1: 递归解法

递归对应树形结构, 最多每个结点可以有 8 个子女, 最坏情况, 时间复杂度为 8 的 n 平方次方, 空间复杂度主要来源于二维数组存储, 空间复杂度为 $O(n^2)$

算法 2: 贪心解法

每次从 8 个点中选择一个点继续探索, 总共的探索次数是确定的, 为 n^2-1 , 因此时间复杂度为 $O(n^2)$, 空间复杂度与算法 1 类似, 为 $O(n^2)$

四、程序实现和实验测试过程

程序实现代码存于 cavalier_traversal 文件
实验测试截图如下：

```
请输入起始坐标，空格隔开
1 3
巡游路径为（回溯）：
+---+---+---+---+---+---+---+---+
| 43| 38| 55| 62| 53| 50| 35| 64|
+---+---+---+---+---+---+---+---+
| 56| 61| 42|  1| 36| 63| 52| 49|
+---+---+---+---+---+---+---+---+
| 39| 44| 37| 54| 51|  2| 31| 34|
+---+---+---+---+---+---+---+---+
| 60| 57| 46| 41| 32| 29| 48|  3|
+---+---+---+---+---+---+---+---+
| 45| 40| 59| 28| 47|  4| 33| 30|
+---+---+---+---+---+---+---+---+
| 58| 27| 16| 19| 22| 11|  8|  5|
+---+---+---+---+---+---+---+---+
| 17| 20| 25| 14|  9|  6| 23| 12|
+---+---+---+---+---+---+---+---+
| 26| 15| 18| 21| 24| 13| 10|  7|
+---+---+---+---+---+---+---+---+

巡游路径为（贪心）（以（3,3）为例：
+---+---+---+---+---+---+---+---+
| 23| 20| 17| 38| 51| 64| 15| 40|
+---+---+---+---+---+---+---+---+
| 18| 37| 22| 57| 16| 39| 52| 63|
+---+---+---+---+---+---+---+---+
| 21| 24| 19| 50| 61| 58| 41| 14|
+---+---+---+---+---+---+---+---+
| 36| 31| 60|  1| 56| 49| 62| 53|
+---+---+---+---+---+---+---+---+
| 25|  2| 47| 32| 59| 54| 13| 42|
+---+---+---+---+---+---+---+---+
| 30| 35| 28| 55| 48| 45| 10|  7|
+---+---+---+---+---+---+---+---+
|  3| 26| 33| 46|  5|  8| 43| 12|
+---+---+---+---+---+---+---+---+
| 34| 29|  4| 27| 44| 11|  6|  9|
+---+---+---+---+---+---+---+---+
```

五、总结

递归算法某些点运行时间较长，而贪心算法某些情况只能求得次优解。