

# 拉丁矩阵问题

2018年5月21日

## 1 题目介绍

### 1.1 问题描述

算法实现题 5-9 拉丁矩阵问题

★问题描述:

现有  $n$  种不同形状的宝石，每种宝石有足够多颗。欲将这些宝石排列成  $m$  行  $n$  列的一个矩阵， $m \leq n$ ，使矩阵中每一行和每一列的宝石都没有相同形状。试设计一个算法，计算出对于给定的  $m$  和  $n$ ，有多少种不同的宝石排列方案。

★编程任务:

对于给定的  $m$  和  $n$ ，计算出不同的宝石排列方案数。

★数据输入:

由文件 input.txt 给出输入数据。第 1 行有 2 个正整数  $m$  和  $n$ ， $0 < m \leq n < 9$ 。

★结果输出:

将计算出的宝石排列方案数输出到文件 output.txt。

输入文件示例

input.txt

3 3

输出文件示例

output.txt

12

## 2 题目分析

本题的解空间为排列树。设  $n$  种宝石编号为  $1, 2, \dots, n$ 。宝石矩阵的第一行从左到右排列为  $1, 2, \dots, n$ ，且第一列从上到下排列为  $1, 2, \dots, m$  的阵列为标准拉丁矩阵。设  $m$  行  $n$  列的标准拉丁矩阵个数为  $L(m, n)$ 。一般情况下， $m$  行  $n$  列的拉丁矩阵个数为  $R(m, n)$ 。本题求  $R(m, n)$ 。而  $R(m, n)$  可表示为:

$$R(m, n) = n!(n-1)!L(m, n)/(n, m)!$$

### 3 算法设计

首先给矩阵进行赋值，每一行赋值都不相同，此时每一行已经不同了，对每一行进行全排列的过程中，每一行一定不会出现相同的情况，所以对每一列进行判断是否存在相同的情况。如果满足每一列都不相同，则此时的矩阵满足条件，方案数加1。

### 4 算法分析

#### 4.1 复杂度

算法的复杂度为 $O(mn!)$ 。