

**课 程 实 验 报 告**

**课程名称： 并行编程原理与实践**

**专业班级： CS1602**

**学 号： U201614545**

**姓 名： 谭胜克**

**指导教师： 金海**

**报告日期： 2019/7/29**

**计算机科学与技术学院**

**目录**

[1 串行算法复杂度分析 1](#_Toc15457923)

[1.1任务描述 1](#_Toc15457924)

[1.2任务分析 1](#_Toc15457925)

[2 分析并行实现的正确性 2](#_Toc15457926)

[2.1任务描述 2](#_Toc15457927)

[2.2任务分析 2](#_Toc15457928)

[3.3实验结果与分析 2](#_Toc15457929)

[3 大数场景分析 3](#_Toc15457930)

[3.1任务描述 3](#_Toc15457931)

[3.2实验方案 3](#_Toc15457932)

[4 并行优化方案设计 4](#_Toc15457933)

[4.1任务描述 4](#_Toc15457934)

[4.2实验方案 4](#_Toc15457935)

[4.3实验结果与分析 4](#_Toc15457936)

# 1 串行算法复杂度分析

## 1.1任务描述

分析串行算法的复杂度，并设计测试案例予以证明。

## 1.2任务分析

设矩阵规模为n\*n。在此条件下分析串行算法，构造黑色方块链表的时候，需要遍历5次矩阵5\*n\*n次，在枚举方案进行尝试的时候，每一种方案开始之前都需要对当前矩阵进行复制，共13次遍历，13\*n\*n次。在黑方块周围放置灯泡的时候需要点亮相应的方块，最坏情况整行整列都需要遍历，2n次。接下来是白块的处理，没轮开始前，寻找空白位置平均需要便利半个矩阵，0.5\*n\*n次，之后是检测函数，该函数第一次遍历，检测空白位置，一次遍历n\*n次。第二次遍历检测黑块周围灯泡数量，同时遇到黑色方块，则进行4次操作判断灯泡数量。

总体来说，整个算法的时间复杂度应该在O(n2)级别。由于存储矩阵的空间为n\*n，所以空间复杂度也在O(n2)级别。

# 2 分析并行实现的正确性

## 2.1任务描述

在“akari回溯”中实现的多种并行实现中，挑选一种并分析其正确性，并请给出方案确保并行实现的正确性。

## 2.2任务分析

从算法的又穷性以及循环不变性给出并行实现的正确性证明。

对于一个给定的矩阵，最多经过有限次的n2的遍历操作即可给出最终的结果，究竟是有解或者无解，所以算法是有穷的。

对于每一个子线程，其所应该处理的是整个解空间数的一个分支，所以在运算范围上不会出现覆盖，最后都经由统一的检验函数对每一个子线程的处理结果不不断的进行检测，所以只要存在结果，必定能够得出。

## 3.3实验结果与分析

如图3.1所示，7项测试均通过，也证明了并行算法的正确性。



图3.1 并行实现平台测试结果

# 3 大数场景分析

## 3.1任务描述

1. 设计大数计算场景下”akari回溯”的测试用例并分别使用串行和并行的方式实现；
2. 分析并行实现的加速比；
3. 分析并行实现加速比的正确性；

## 3.2实验方案

对于大数场景，对于并行粒度的划分十分关键。在数据规模变大时，考虑到线程的开销问题，不能够使并行粒度太大；同时为了能够充分发挥并行计算的优势，并行粒度也不能太小。

所以相比较而言当数据规模特别大的时候，选用当且仅当解空间树结点第一次分裂的时候产生新任务。

# 4 并行优化方案设计

## 4.1任务描述

1. 设计大数场景并行实现的优化方案；
2. 测试并分析优化方案的加速比与正确性；

## 4.2实验方案

将实验2中的并行回溯法进行改造，改为仅在第一个任务开始时新建线程采用并行化处理，其余均串行处理。

## 4.3实验结果与分析

由图4.1、4.2，可知改进后的并行程序加速比为10。



图4.1 串行算法执行时间

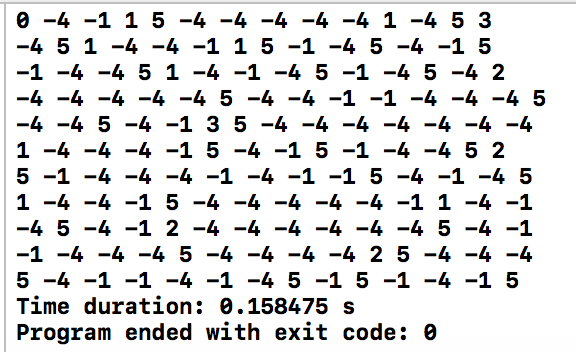


图4.2 并行算法执行时间