# Lab1 test report

### 1.实验概述

**1.1输入**

程序在控制台接收输入的测试文件，测试文件在当前版本的代码的目录下，包括test1,test10…test1000，,文件中包含一个到多个的数独题，并且这些数独按照一定的格式存储。

**1.2输出**

程序将测试文件的数独的解按照与输入相应的顺序写到输出文件outfile中。

**1.3 Sudoku算法**

实验提供了四个算法：BASIC,DANCE,MINA 和 MINAC。本实验选择使用BASIC算法。

**1.4实验环境**

本次实验有3个不同的实验环境，分别是：

（1）yudan：Linux内核版本为4.15.0-72-generic；1GB内存；CPU型号为Intel®Core™i5-7200U [CPU@2.50GHz](mailto:CPU@2.50GHz)，共一个物理CPU，每个物理CPU有4个物理核心；不使用超线程技术。

（2）tjc：

（3）服务器：

在测试时会说明不同的测试环境

**1.5 代码版本**

本次实验中使用了三份代码：

（1）easy\_version，静态分配各个线程任务后各自执行等

（2）Basic\_version，使用任务队列分配任务等

（3）Advanced\_version，在Basic\_version版本基础上，使用一个专门的线程接受输入，并且可以接受任何数量的输入文件等。

### 2.性能测试

本次实验的性能分析将比较

1. 实验中实现的不同版本的代码在相同的测试环境、不同的输入文件以及同样使用单线程进行求解的性能差别；
2. 实验中实现的某一版本的代码在相同的测试环境下、相同的文件输入以及不同的线程数量的性能差别；

③ 实验中实现的某一代码在不同的测试环境下、相同的文件输入以及相同的线程数量的性能差别。

**2.1不同版本的代码**

代码版本：Advanced\_version，Basic\_version以及easy\_version

输入：从test1到test1000，使用单线程求解

测试环境：服务器

这三个版本的代码实现了相同的功能，但是在时间上的开销是不同的，随着问题规模的不断增大，不同版本的代码时间上的开销的差距会更加明显的显现。本次实验对这些代码进行时间开销上的对比，分别按照输入文test1,test10,test20,

test30,test50,test100,test200,test500,test100对代码版本Advanced\_version，Basic\_version以及easy\_version进行测试。

结果分析：  
从上图中可以看出当输入文件的数独数量不断增加，Advanced\_version，

Basic\_version以及easy\_version的时间开销差距越来越明显。

在test1000时easy\_version比Basic\_version多花了将近14秒的时间，这是

因为basic比easy版本实现代码更多，并且随着输入文件数独问题数量的增多，额外的开销的影响就体现出来了，所以最终会造成上图这种情况。

**2.2不同线程数**

代码版本为Basic\_version

输入：test100

测试环境：yudan

针对Basic\_version版本的代码在不同的线程数的使用情况下，测试其时间性能，线程的数量为1,2,4,8,……1024，使用某个数量的线程对所提供的输入文件test100中的数独题进行求解，并且将结果写入输出文件outfile中，测量则个过程的时间开销。

结果分析：

在64个线程时，时间开销增加

**2.3不同硬件环境**

实验使用Basic\_version版本的代码在两个不同的硬件环境下，针对输入文件test100进行求解测试，sudoku\_solve线程数量从1开始增加, 测量性能。

测试环境: tjc、yudan

如下图所示为在不同硬件环境下的测试结果:

结果分析：

从图中可以看出，