**数据结构实验报告**

姓名： 粟锦 学号：U201817053 班级：软工1803班

**实验一 求整数和、铺地板和Hanoi塔等问题的求解**

一、实验描述

1、用C语言编程实现求整数平方和，铺地板以及Hanoi塔等问题的求解。

2、在程序中加入clock ()来计算求解时间。

3、使用不同的输入值得到对应的时间值。

4、分析算法的时间复杂度并与测量结果进行比较。

5、如果存在差异，解释原因。

二、求解思路

1.求整数平方和问题：

利用for循环进行叠加求解。

输入100、200、300、400、500、600、700共7组数据。

2.铺地板问题：

利用迭代的方式进行求解，有f(n) = f(n-1)+ f(n-2)，且f(0)=1，f(1)=1。

输入100、200、300、400、500、600、700共7组数据。

3.Hanoi塔问题

利用递归调用的方式进行求解，得到f(n)=2f(n-1)+1，f(1)=1的递推公式。

输入20、21、22、23、24、25、26共7组数据。

三、实验实现过程

1.求整数平方和

（1）定义int型变量n并通过scanf\_s()函数对变量n赋值，把n作为求和函数的参数。

（2）定义clock\_t类型的变量start和end并调用clock（）函数记录求值函数的执行时间。

（3）将求值函数循环100000次，计算总共的执行时间。一共记录5次取平均值。

（4）改变n的值，重复（3）。

（5）绘制图表。

2.铺地板

（1）定义int型变量n并通过scanf\_s()函数对变量n赋值，把n作为求和函数的参数。

（2）定义clock\_t类型的变量start和end并调用clock（）函数记录求值函数的执行时间。

（3）将求值函数循环100000次，计算总共的执行时间。一共记录5次取平均值。

（4）改变n的值，重复（3）。

（5）绘制图表。

3.Hanoi塔问题

（1）定义int型变量n并通过scanf\_s()函数对变量n赋值，把n作为求和函数的参数。

（2）定义clock\_t类型的变量start和end并调用clock（）函数记录求值函数的执行时间。

（3）计算执行时间，一共记录5次取平均值。

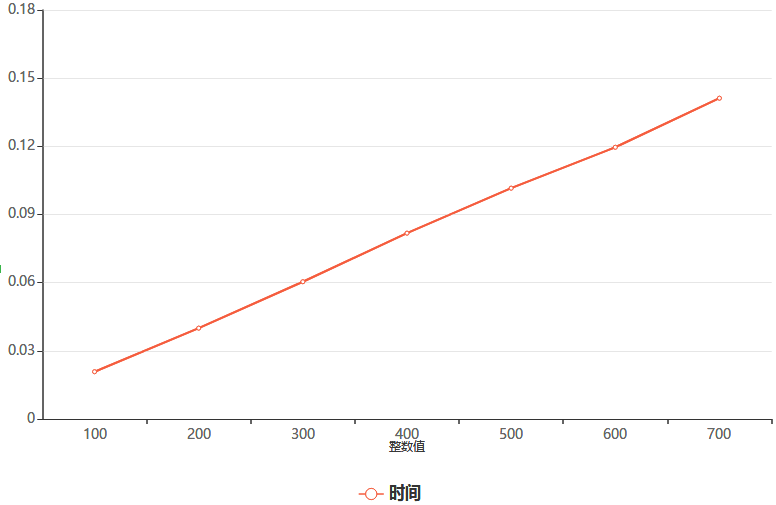
（4）改变n的值，重复（3）。

（5）绘制图表。

四、实验结果

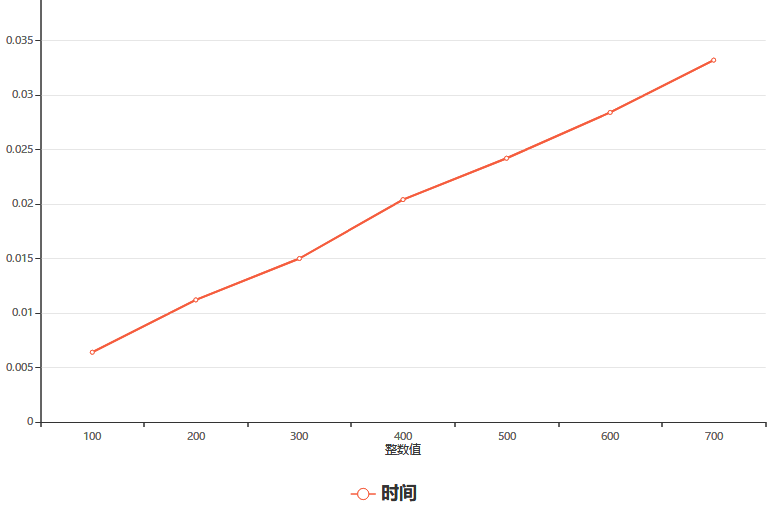
1.求整数平方和

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 整数值 | 100 | 200 | 300 | 400 | 500 | 600 | 700 |
| 时间 | 0.0208 | 0.0400 | 0.0604 | 0.0818 | 0.1016 | 0.1196 | 0.1412 |

2

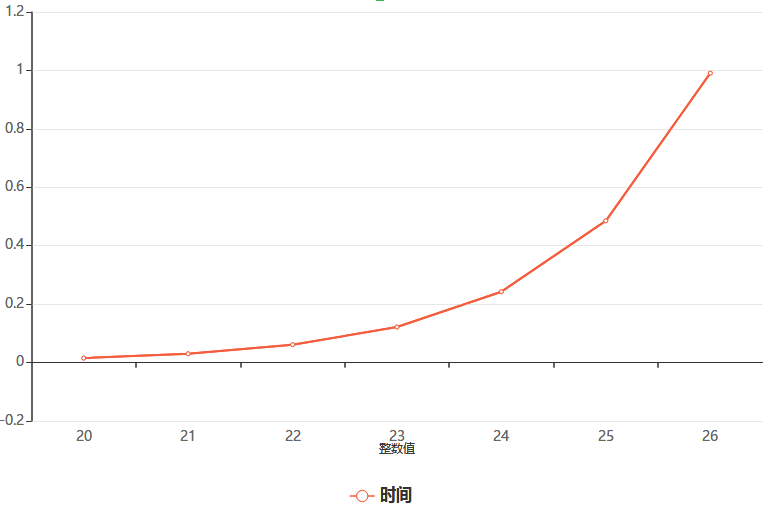
1. 铺地板

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 整数值 | 100 | 200 | 300 | 400 | 500 | 600 | 700 |
| 时间 | 0.0064 | 0.0112 | 0.0150 | 0.0204 | 0.0242 | 0.0284 | 0.0332 |



3.Hanoi塔问题

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 整数值 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 时间 | 0.0154 | 0.0302 | 0.0612 | 0.1216 | 0.2426 | 0.4852 | 0.9908 |



五、实验结论

1.求整数平方和

（1）算法时间复杂度：

应用for循环进行一次遍历求得和，其复杂度T(n)=O(n)。

（2）与测量结果进行比对

求整数平方和问题最后得到的结果曲线为一条近似直线，n与时间t成线性关系，与算法时间复杂度分析所得的T(n)=O(n)吻合。

1. 铺地板

（1）算法时间复杂度：

应用for循环迭代进行一次遍历求得和，其复杂度T(n)=O(n)。

（2）与测量结果进行比对

铺地板问题最后得到的结果曲线为一条近似直线，n与时间t成线性关系，与算法时间复杂度分析所得的T(n)=O(n)吻合。

3.Hanoi塔问题

1. 算法时间复杂度：

由递推公式T(n) = 2T(n-1)+1可得，时间复杂度T(n)=O(2^n)。

（2）与测量结果进行比对

由图可知，Hanoi塔问题最后得到的结果曲线近似为一条指数函数线，n与时间t成指数关系，与算法时间复杂度分析所得的T(n)=O(2^n)吻合。

由测试结果与理论分析可知，时间复杂度与运行时间之间的关系与理论较为吻合。存在差异的原因可能是系统CPU的内存分配在不同时间有所改变造成的误差。