**数据结构**

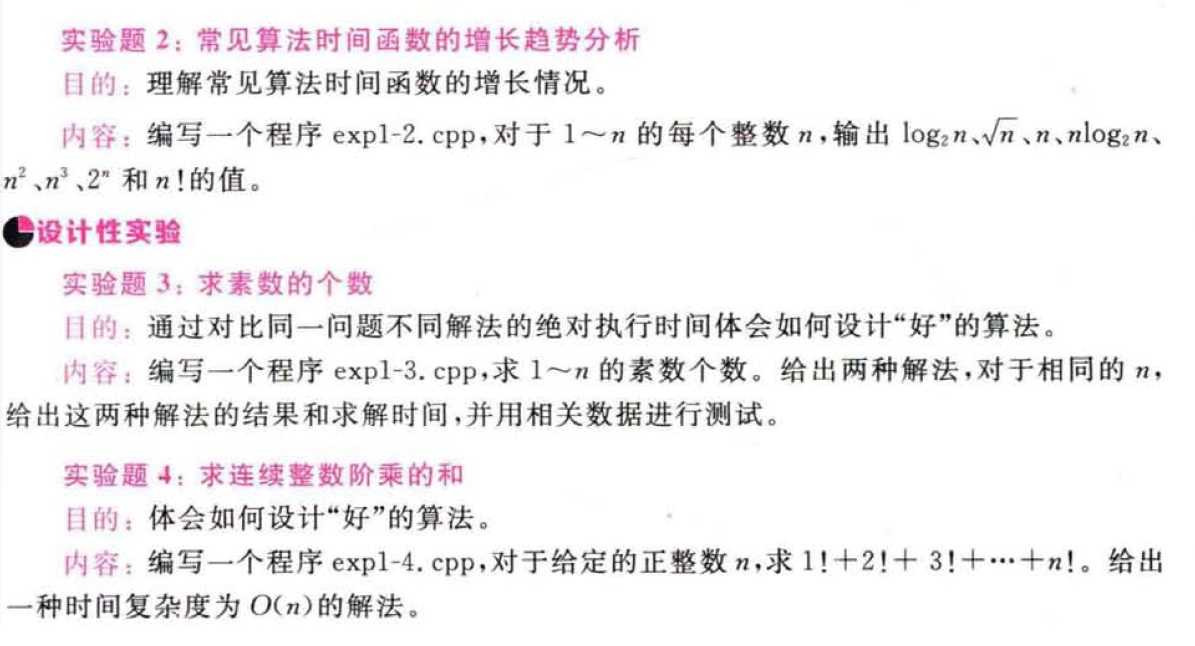
2022

**实 验 报 告**

|  |  |
| --- | --- |
| 实验项目名称： | 数据结构实验一 |
| 班级： | 21级9班 |
| 学号 | 2021302121185 |
| 姓名： | 陈聪睿 |
| 指导教师： | 董红斌 |
| 实验时间： | 2022.3.17 |

**实验一： 常见算法时间函数的增长趋势分析**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

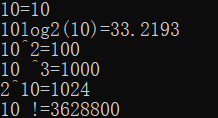
**三、实验步骤及思路**

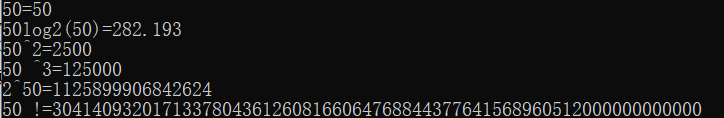
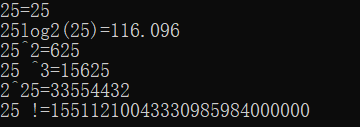
（1）题目分析与主要步骤

需要使用log2（），sqrt（），pow（）等公式，需要引入<math.h>库。同时n！的求法需用大数运算，可以利用开辟数组存储显示，并封装成函数，利于主函数代码的阅读。

（2）实验具体步骤截图

测试数据：n=10,n=25,n=50(只列出i=n时的最终结果)





**四、实验结果及分析**

可以看出，时间函数按照如题目要求列出的运算，从小到大依次变大，并且按照其顺序，随着n的增长而增长越来越快。

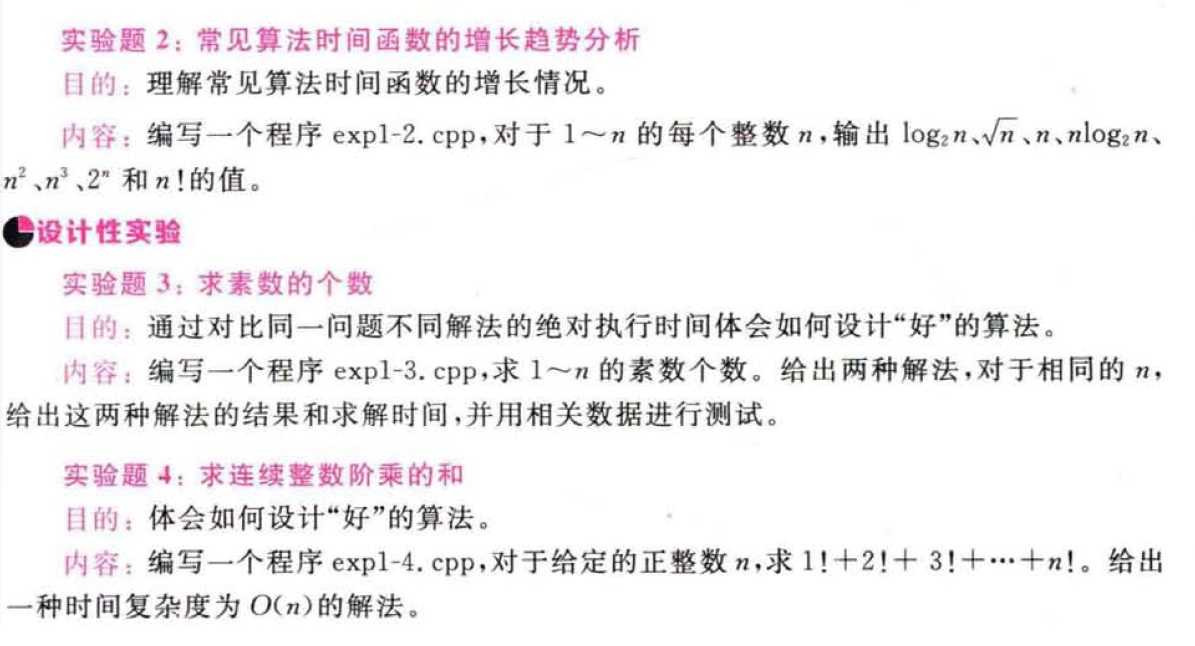
**五、总结**

数据到最终，数据量会变得十分大，容易溢出，需要灵活的变化存储方式。

同时对于n！这种大运算，需要使用数组存储每一位，尔后依次输出，才能保证其不会溢出。

**实验二： 求连续整数阶乘的和**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

可以使用已经算过的结果，将前一次的阶乘数据乘本次数据即可得到本次阶乘的结果，即n！=n\*（n-1）！，尔后将其与前一次的结果相加，得到现在的结果，即result[n]=result[n-1]+ n\*（n-1）！。

这样，只需要循环一次，就可以得到最终结果。

（2）实验具体步骤截图

测试数据：n=5,n=10,n=15，n=20









**四、实验结果及分析**

我们可以观察到前一项与后一项的关系，即result[n]=result[n-1]+ n\*（n-1）！。通过使用这个公式，我们就可以只是用一次循环，就可以使得问题得以简单解决，并且将其时间复杂度控制在O（n）。

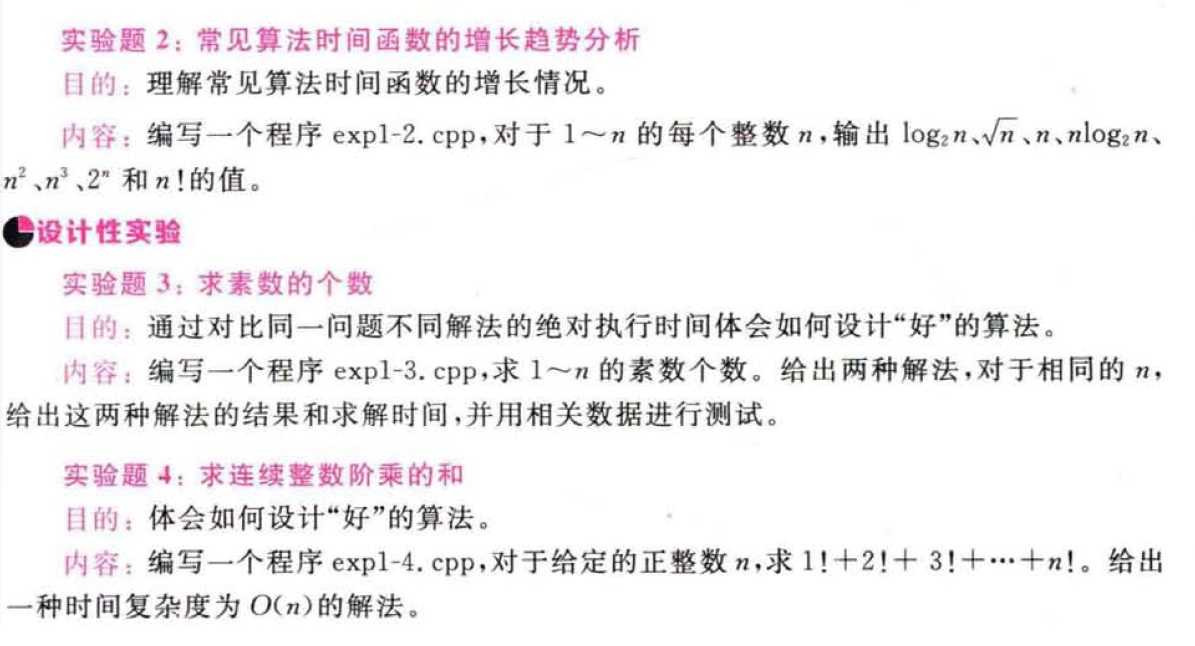
**五、总结**

合理使用之前运算过的数据，并且学会减少循环的使用次数，对于我们减少时间复杂度有很大的用处。

但是到最终，数据量会变得十分大，容易溢出，仍需要使用大数运算的方法避免，仍需改进。

**实验三： 求素数个数问题**

**一、实验要求**



**二、实验环境**

**Visual studio 2019/ Windows平台**

**三、实验步骤及思路**

（1）题目分析与主要步骤

第一次求素数个数（函数设为prime1），可以将本次循环所要测试的数据k与1~k之间的数进行取模运算，如果取模为零，证明这个数为合数，将其个数cnt记录下来，最终结果可用总个数n减去cnt得到。

第二次求素数个数（函数设为prime2），可以记录第一次遇到的素数，并将其的倍数（小于规定的最大数）记录为合数，最后可以在循环验证的过程中直接记录素数的个数cnt。

（2）实验具体步骤截图

测试数据：n=100



**四、实验结果及分析**

两个程序都正确得到了结果，但是其用时并不相同。

经过测试得到：

**Prime1在1~100内，debug模式下，用时1.947s**

**Prime2在1~100内，debug模式下，用时1.437s**

可以看出第二种方法的时间用时最短，其原因是利用了原有数据，减少了每次运算所需的数据，提升了运算效率。

**五、总结**

我们在判断其是否为素数的时候，可以利用已经判断过的素数，这样子可以减少判断的次数，比传统的遍历测试更优。