算法设计与分析实验报告

<table>
<tr><td>班级</td><td></td><td>学号</td><td></td><td>姓名</td><td></td></tr>
<tr><td>实验名称</td><td>算法复杂性分析</td><td>日期</td><td>2023.9.18</td><td>成绩</td><td></td></tr>
<tr><td>实验目的和要求</td><td colspan="5">1. 文件数据读取;
2. 时间复杂性测试;
3. 简单程序编写,理解顺序、分支和循环结构.</td></tr>
<tr><td>实验准备</td><td colspan="5">熟悉编程环境；复习 C/C++语法；</td></tr>
<tr><td>实验内容、实验结果与分析</td><td colspan="5">问题 1：设计一个程序，计算某只股票（或者基金/指数）大约 n(n≥50)个交易日的最大连续上涨(涨幅≥0.5%)天数。要求给出数据说明和来源，以及运行结果。
数据及其来源：
北证 50(899050) - 历史行情 - 股票行情中心 - 搜狐证券 (sohu.com)
2023-5-26 至 2023-9-18 共 80 条数据
程序：
<pre>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main() {

    // *以下打开 csv 文件的代码引用了他人代码并做出部分改变，来源：https://www.cnblogs.com/mengxinayan/p/22-02-read-write-csv-in-c-1.html
    FILE* csvFile;
    const char* filename = "北证 50 股票行情.csv";
    const char* mode = "r";

    fopen_s(&csvFile, filename, mode);   // 读取 CSV 文件
    char line[80];
    double fluctuation[80];   // 涨幅数据储存在 fluctuation 数组中
    int count = 0;
    while (fgets(line, 80, csvFile)) {
        double value = atof(line);   // 将读取的字符串转换为 double
        fluctuation[count] = value;   // 存储到数组中
        count++;
    }
    fclose(csvFile);   // 关闭文件
    // *引用结束
</pre>
</td></tr>
</table>

```
    int max = 0;            // 最大连续上涨天数
    int current = 0;            // 当前连续上涨天数

    for (int i = 0; i < 80; i++) {
        if (fluctuation[i] >= 0.05) {
            current++;
        }
        else {
            // 如果涨跌幅不大于等于 0.5%，重置当前连续上涨天数
            current = 0;
        }

        // 更新最大连续上涨天数
        if (current > max) {
            max = current;
        }
    }
    printf("最大连续上涨不小于 0.5%%的天数：%d\n", max);

    return 0;
}
```

运行结果：

```
最大连续上涨不小于 0.5%的天数：6
```

问题 2：把如下 3 个程序补充完整并测试，分别写出 N=500，1000，1500，2000，2500，3000 的运行结果和运行时间，用 Excel 画出 3 条折线图（同一坐标系下）

程序 1，

```
{    int i,N=500;
     Long J=1;
     Double S=1;
     For(i=0;i<N;i++){
      J=J+1;
      S=S + 1.0/J;
     }
     Printf("S=%lf",S);
}
```

程序 2，

```
{    int i1,i2,N=500;
     Long J=1;
```

```
    Double S=1;
    For(i2=0;i2<N;i2++){
      For(i1=0;i1<N;i1++){
       J=J+1;
       S=S + 1.0/J;
      }
    }
    Printf("S=%lf",S);
}
```

程序 3,

```
{   int i1, i2,i3,N=500;
    Long J=1;
    Double S=1;
    For(i3=0;i3<N;i3++){
      For(i2=0;i2<N;i2++){
        For(i1=0;i1<N;i1++){
         J=J+1;
         S=S + 1.0/J;
        }
      }
    }
    Printf("S=%lf",S);
}
```

实验结果与分析：

程序 1：

```c
#include <stdio.h>
#include <time.h>

int main(void){

    clock_t start, end;
    double time_used;
    start = clock();        //记录程序开始时间

    int i, N = 500;
    long J = 1;
    double S = 1;
    for(i = 0; i < N; i++) {
        J = J + 1;
        S = S + 1.0 / J;
    }
    printf("S=%lf\n", S);
```

```
    end = clock();      //记录程序结束时间
    time_used = (double)(end - start) / CLOCKS_PER_SEC;
    printf("程序运行时间：%1f 秒 \n", time_used);

    return 0;
}
```

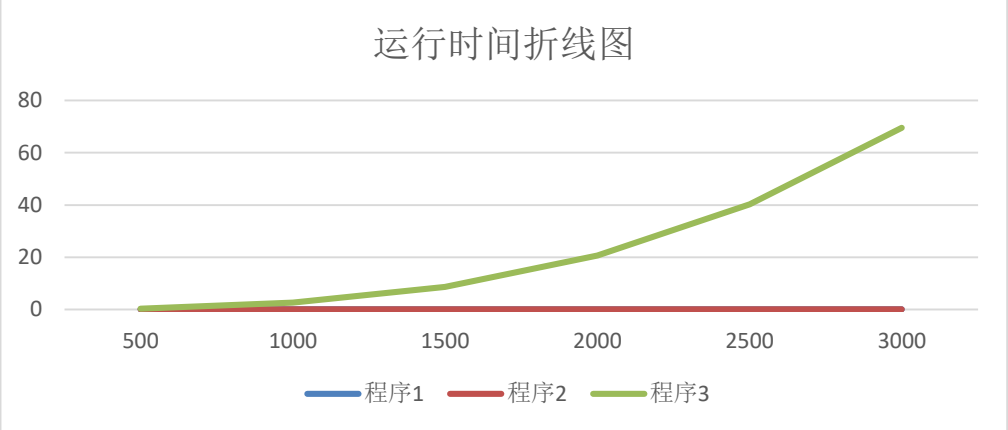
程序 2：

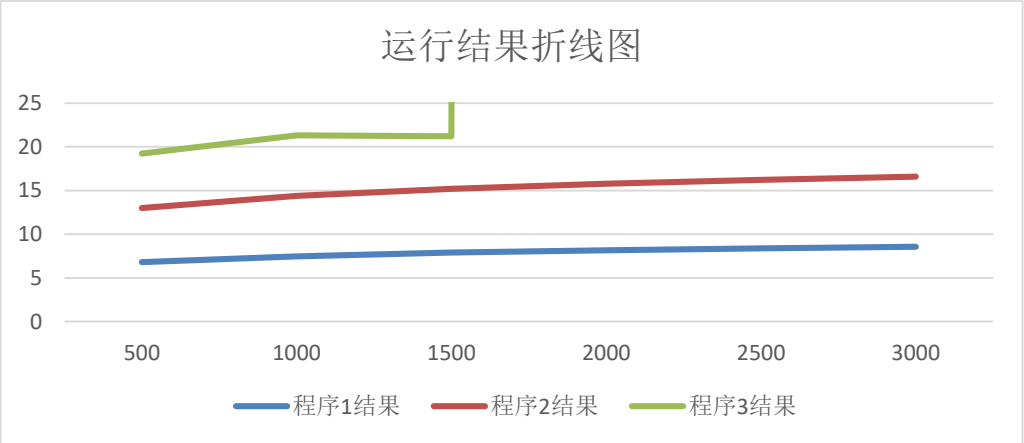
```
#include <stdio.h>
#include <time.h>

int main(void){

    clock_t start, end;
    double time_used;
    start = clock();        //记录程序开始时间

    int i1, i2, N = 500;
    long J = 1;
    double S = 1;

    for (i2 = 0; i2 < N; i2++) {
        for (i1 = 0; i1 < N; i1++) {
            J = J + 1;
            S = S + 1.0 / J;
        }
    }
    printf("S=%lf\n", S);

    end = clock();      //记录程序结束时间
    time_used = (double)(end - start) / CLOCKS_PER_SEC;
    printf("程序运行时间：%.6f 秒 \n", time_used);

    return 0;
}
```

程序 3：

```
#include <stdio.h>
#include <time.h>

int main(void){
```

```
    clock_t start, end;
    double time_used;
    start = clock();        //记录程序开始时间

    int i1, i2, i3, N = 500;
    long J = 1;
    double S = 1;
    for (i3 = 0; i3 < N; i3++) {
        for (i2 = 0; i2 < N; i2++) {
            for (i1 = 0; i1 < N; i1++) {
                J = J + 1;
                S = S + 1.0 / J;
            }
        }
    }
    printf("S=%lf\n", S);

    end = clock();      //记录程序结束时间
    time_used = (double)(end - start) / CLOCKS_PER_SEC;
    printf("程序运行时间: %.6f 秒 \n", time_used);

    return 0;
}
```





	500	1000	1500	2000	2500	3000
程序 1	0	0	0	0	0	0
1 结果	6.794819	7.48647	7.891436	8.178868	8.401862	8.584083
程序 2	0	0.002	0.006	0.01	0.016	0.023
2 结果	13.00644	14.39273	15.20366	15.779021	16.225308	16.589951
程序 3	0.322	2.576	8.686	20.622	40.27	69.538
3 结果	19.22104	21.30048	21.21706	9.00E+307	9.00E+307	9.00E+307

问题 3：输入正整数 n；把从 1 至 n 的 n 个整数中排在第奇数序号的数去掉，对于剩余的数重复这个操作，直到剩下一个数，输出这个数. 填表

n	34	145	256	367	478	589
剩余数	32	128	256	256	256	512
Nc	5	7	8	8	8	9

```
int findNumber(int n) {
    if (n == 1) {
        return 1; // 如果 n 为 1，直接返回 1
    } else {
        // 递归调用，去掉奇数序号的数
        return 2 * findNumber(n / 2);
    }
}
```

总结与讨论	

注：完成作业后，把电子版提交至 课堂派；引用他人的程序注明来源；

作业名称：学号姓名实验 01 时间复杂性