作业 HW0 实验报告

姓名：何正潇 学号：1950095 日期：2021 年 12 月 24 日

1. 涉及数据结构和相关背景

本单元主要是 oj 系统的熟悉与测试，练习了一些基本算法思想。

2. 实验内容

2.1 最大子段和（题目名字）

2.1.1 问题描述

给定K个整数组成的序列{ N_1, N_2, ..., N_K }，“连续子段”被定义为{ N_i, N_{i+1}, ..., N_j }，其中 $1 \le i \le j \le K$。“最大子段和”则被定义为所有连续子段元素的和中最大者。例如给定序列{ -2, 11, -4, 13, -5, -2 }，其连续子段{ 11, -4, 13 }有最大的和20。现要求你编写程序，计算给定整数序列的最大子段和。

2.1.2 基本要求

第1行输入1个正整数K，表示序列中的元素个数
第2行输入K个整数ai，空格分割，表示序列中的元素值
输出是最大字段和的值

2.1.3 数据结构设计

没有使用典型数据结构

2.1.4 功能说明（函数、类）

```
/*1950095 大数据 何正潇*/
#define _CRT_SECURE_NO_WARNINGS
#include <string.h>
#include <string>
using std::string;
#include <iostream>
#define num 60000
using namespace std;
int main()
{
    int number;
    cin >> number;
    int* temp = NULL;
    temp = new int[number];
    for (int i = 0; i < number; i++)
```

```
        {
            cin >> temp[i];
        }
        int max = 0;
        int sum = 0;
        for (int i = 0; i < number; i++)
        {
            if (sum <= 0 && temp[i] >= 0)
            {
                sum = temp[i];
                if (sum >= max)
                    max - sum;
            }
            else if (sum >= 0)
            {
                sum += temp[i];
                if (sum >= max)
                    max = sum;
            }
            else
                continue;
        }
        cout << max;
        return 0;
    }
```

2.1.5 调试分析（遇到的问题和解决方法）

没有进行调试，基本一次就 pass

2.1.6 总结和体会

这题更多来说使用了一个简单的动态规划思想。主要通过这道题熟悉了 oj 系统。

1-2 大整数乘法

1.2.1 问题描述

用过Python的同学都会知道，其自带的长整型类型可以计算几百位的乘法。我们希望同学们能够自己实现这样一个长整型乘法，本题求两个不超过200位的非负整数的积。

1.2.2 基本要求

输入共包含n+1行，第1行1个正整数n，表示要计算的n组乘法
接下来n行，每行两个非负整数a 和 b，空格分割，表示进行乘法运算的两个数

1.2.3 数据结构设计

没有使用典型数据结构

1.2.4 功能说明（函数、类）

主要使用的分治思想，用一位和整个数字进行运算，最后得到结果。

```
    for (int temp = 0; temp < number; temp++)
    {
        char s1[205], s2[205];
        int str1[205], str2[205];
        int len1, len2, i;
        scanf("%s %s", s1, s2);
        len1 = strlen(s1), len2 = strlen(s2);
        memset(str1, 0, 205);//初始化 0
        memset(str2, 0, 205);
        int len = 0;
        for (i = 0; i < len1; ++i)
            str1[i] = s1[len1 - 1 - i] - '0';
        for (i = 0; i < len2; ++i)
            str2[i] = s2[len2 - 1 - i] - '0';

        for (i = 0; i < len1; ++i)
        {
```

```
            int b = 0; //每遍历完数组 a 的一个数，进位 b 都要初始化为 0
            int j = 0;
            if (len2 == 1 && str2[0] == 0)
            {
                length[temp] = 1;
                result[temp][0] = 0;
                break;
            }
            else
            {
                for (j = 0; j < len2 || b; ++j)//当 str[j]没遍历完，或者最高位满十需要进位，进位不为 0
                {
                    if (j < len2)
                    {
                        int t = result[temp][i + j] + str1[i] * str2[j] + b;
                        result[temp][i + j] = t % 10; //余数就是该 ans[i+j]位置的数
                        b = t / 10;
                    }
```

```
                else
                {
                    result[temp][i + j] = b;
                    b = 0;
                }
            }
            length[temp] = i + j;
        }
    }
}
```

1.2.5 调试分析（遇到的问题和解决方法）

主要的难点在于分治想法的算法设计，当然其实我是先做级数相加的，做这题时已经没什么难度了。

1.2.6 总结和体会

通过这道题主要是熟悉分治思想，同时熟悉 oj 系统的使用和报告的撰写格式。Hw0 的题目更多只是练习和体会。