**云南大学2021年秋季学期信息学院**

**计算机科学与技术/智能科学与技术专业2020级**

**《数据结构实验》期末大作业报告**

**满分：100分 任课教师： 肖清**

**学院： 信息学院 专业： 计算机科学与技术**

**学号：20201060330 姓名： 胡诚皓 成绩：＿\_\_＿＿＿**

1. 问题描述与设计要求

给定任意n 个不同的正整数a1, …, an 及任意一个正整数B，查找这n 个正整数中所有两两之和等于正整数B的整数对。

编写一个查找整数对的程序，要求如下：

（1）算法的时间复杂度为O(nlogn)；

（2）程序具有操作界面；

（3）可以测试多组数据，每组测试数据通过键盘输入，包括任意n 个不同的正整数a1, …, an 及任意一个正整数B；

（4）将每组测试数据及查找结果保存到IntegerPair.txt 文件中，格式如下：

第一组测试数据

9 个不同的正整数：3 6 2 9 1 12 14 21 16

正整数B：23

查找结果：有2 对整数对：2 21， 9 14

**二、分析与设计**

**（包括数据结构设计、算法设计、功能模块、程序流程图等）**

**数据结构**

定义了一个通用的模板结构体Pair，用于存储相同类型的成对的数据；使用typedef定义了intPair作为存储int类型的数据对的结构。为了便于使用，定义了包含intPair的增强数组intPairArr，其中包含了intPair类型的指针arr用于存放intPair类型的数组、size用于存储已经记录的intPair的个数。

使用了实验九中的快速排序，将dataType从char改为int即可。

**算法设计**

void quickSort(dataType [], int left, int right)

快速排序函数，与实验九中的一致，不再赘述。

void solve(int \*a, int N, int B, int num)

a表示当前要操作的正整数序列，N表示这个正整数序列中的元素个数，B表示要求的正整数对的和，当前操作的组号用num表示，即当前处理的为第num组正整数序列。在实际查找过程中，采用的是类似滑动窗口的思想。

①对数组a进行升序的快速排序操作

②初始化两个left、right两个“指针”，分别指向序列中的第一个和最后一个元素

③left在right的左侧则转到④，否则转到⑤

④若left、right所指的两个正整数的和为B，则将它们加入res中，并将left、right分别右、左移以推动循环；若left、right所指的两个正整数的和比B小，则将left右移，尝试将加和“略微”变大一些；若left、right所指的两个正整数的和比B大，则将right左移，尝试尝试将加和“略微”变小一些。回到③

⑤通过重定向输出流，调用display按模板分别将结果输出到文件IntegerPair.txt和控制台（便于观察）

int main()

①定义变量。使用fopen函数使用“w+”模式打开目标文件，若文件已经存在，会清空文件中原先的内容，否则会创建一个新的空文件

②先读入需要处理的正整数的个数n，申请n个int类型的空间给指针a。再需要求的两数之和B与正整数序列中的各个数

③调用solve函数解决问题

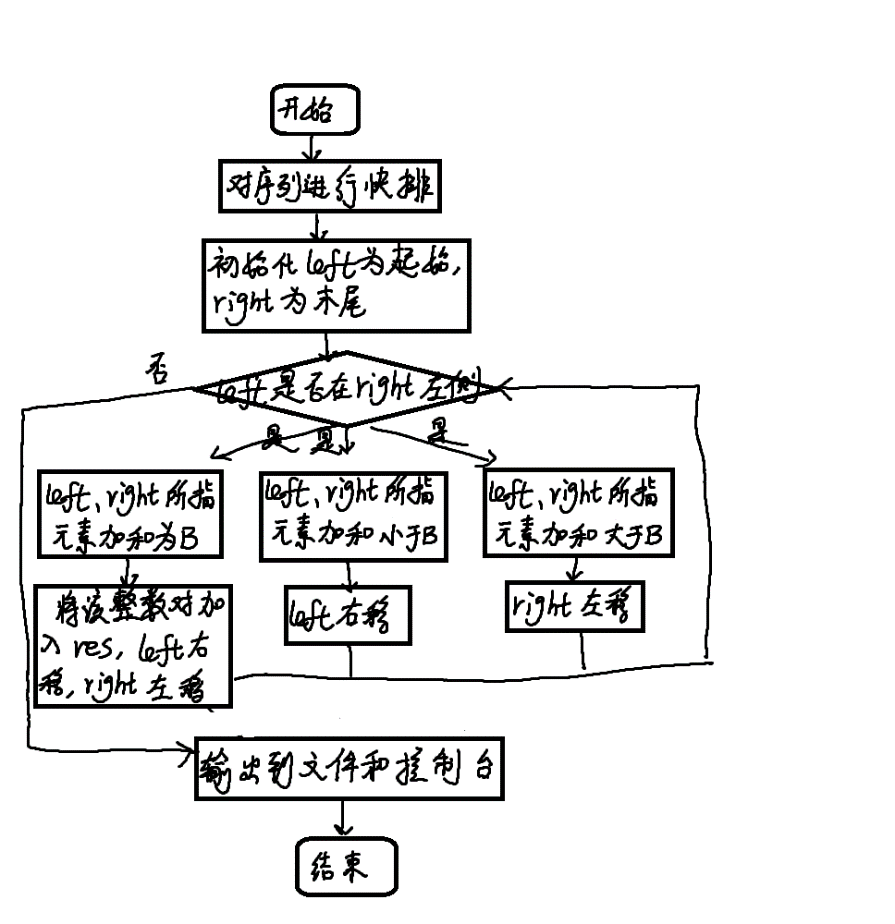
④清除申请的空间

整个程序中，快速排序时间复杂度为；读入正整数序列时，时间复杂度为；在找正整数对时，每个元素遍历一遍，时间复杂度为。因此，最终的时间复杂度为。

**功能模块**

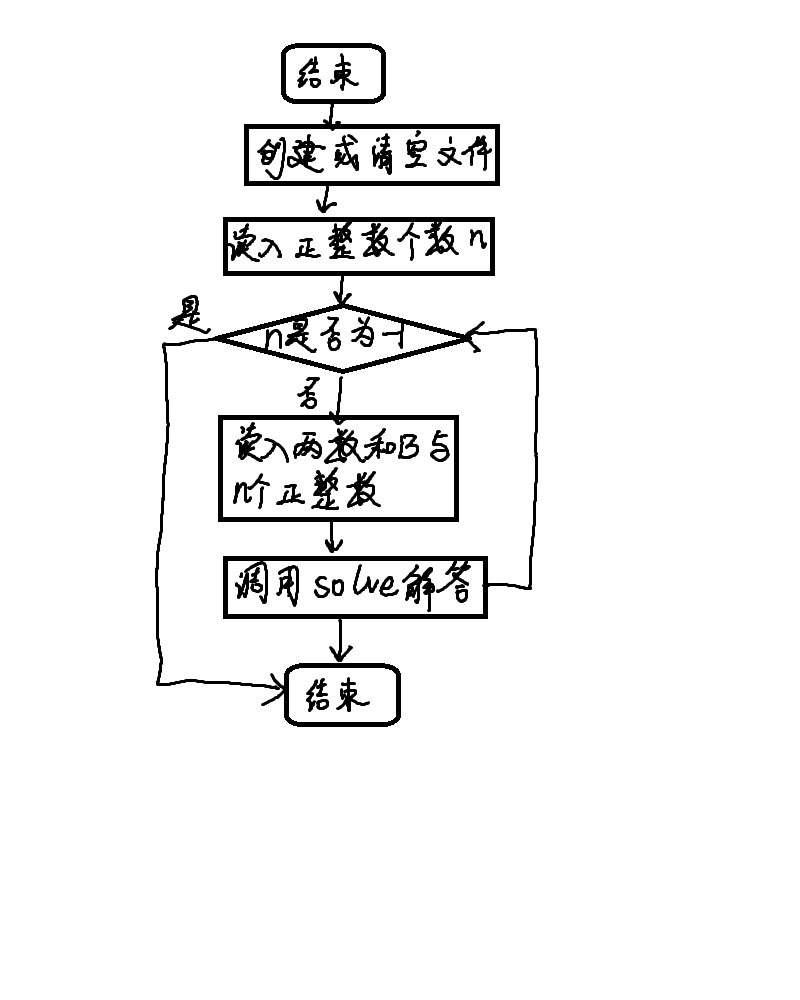
程序主要分为两大模块，一是用于解决问题的核心模块，即solve函数；二是控制输出格式的输出模块，即display函数。

**程序流程图**

****

solve函数示意图

main函数示意图



**三、核心程序代码**

**（给出必要注释）**

（限于篇幅，只附上了解决问题的核心代码

void solve(int \*a, int N, int B, int num)

{

//先对数组a进行快排

quickSort(a, 0, N - 1);

int left = 0, right = N - 1;

int resNum=0;

intPairArr res;//用于存放找到的正整数对

res.arr = new intPair[N/2];//最多可能有N/2个整数对

while (left < right)

{

if (a[left]+a[right] == B)//找到一对

{

res.arr[resNum].first = a[left];

res.arr[resNum].second = a[right];

resNum++;

res.size++;

//推进查找

left++;

right--;

}

else if (a[left]+a[right] < B)//不够大

{

left++;

}

else//太大了

{

right--;

}

}

//重定向输出到文件并输出

freopen("./IntegerPair.txt", "a", stdout);

display(num, N, a, B, res);

//重定向到控制台，再输出一次，以便观察

freopen("CON", "w", stdout);

display(num, N, a, B, res);

}

void display(int num, int N, int \*a, int B, intPairArr ans)

{

cout << "第" << num << "组测试数据" << endl;

cout << N << "个不同的正整数：";

for (int i = 0; i < N; i++)

{

cout << a[i] << " ";

}

cout << endl << "正整数B：" << B << endl;

cout << "查找结果：有" << ans.size << "对整数对：";

//无正整数对就不输出

if (ans.size != 0)

{

cout << ans.arr[0].first << " " << ans.arr[0].second;

for (int i = 1; i < ans.size; i++)

{

cout << ", " << ans.arr[i].first << " " << ans.arr[i].second;

}

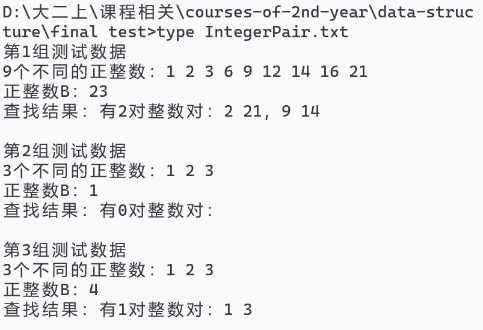
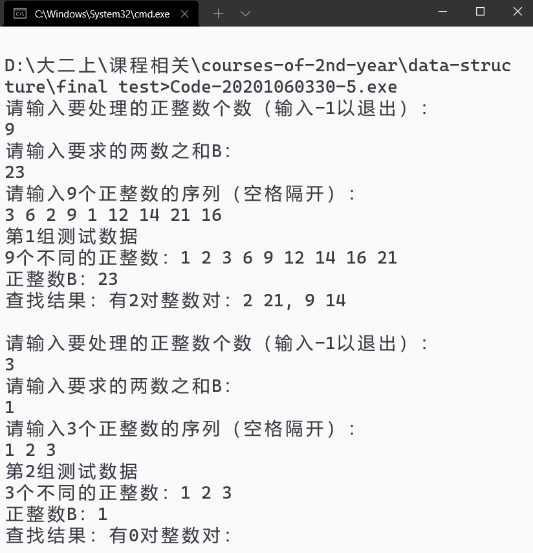
}

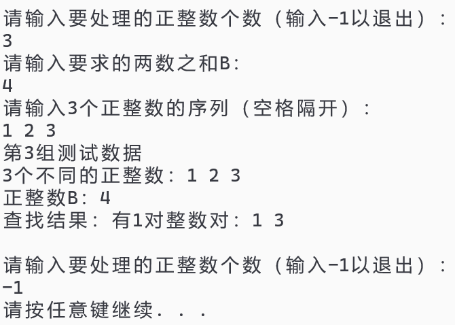
cout << endl << endl;

}

**四、测试与结果**

**（给出测试用例及测试结果）**





**五、用户使用说明**

IntegerPair.txt文件将会保存在exe文件所在的文件夹中，若打开txt发现是乱码，请将txt文件另存为ANSI编码（Windows中文语言系统下默认为GBK）。

