

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： _____

一、问题描述与设计要求

给定任意 n 个不同的正整数 a_1, \dots, a_n 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 $O(n \log n)$ ；

(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 a_1, \dots, a_n 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

③`left` 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

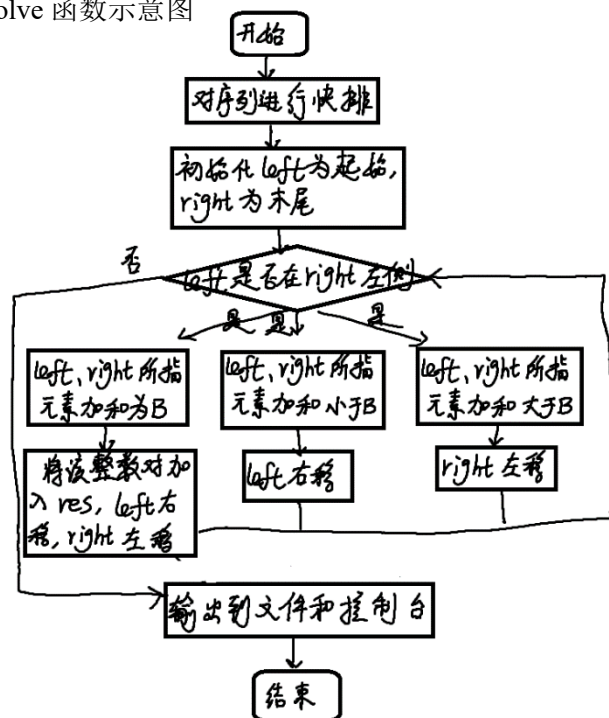
整个程序中，快速排序时间复杂度为 $O(n \log n)$ ；读入正整数序列时，时间复杂度为 $O(n)$ 在找正整数对时，每个元素遍历一遍，时间复杂度为 $O(n)$ 。因此，最终的时间复杂度为 $O(n \log n)$ 。

功能模块

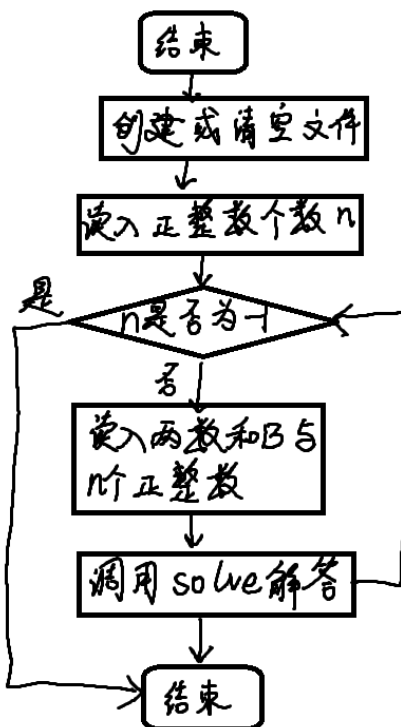
程序主要分为两大模块，一是用于解决问题的核心模块，即 `solve` 函数；二是控制输出格式的模块，即 `display` 函数。

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）

```

D:\大二上\课程相关\courses-of-2nd-year\data-structure\final test>Code-20201060330-5.exe
请输入要处理的正整数个数（输入-1以退出）：
9
请输入要求的两数之和B：
23
请输入9个正整数的序列（空格隔开）：
3 6 2 9 1 12 14 21 16
第1组测试数据
9个不同的正整数： 1 2 3 6 9 12 14 16 21
正整数B： 23
查找结果： 有2对整数对： 2 21, 9 14

请输入要处理的正整数个数（输入-1以退出）：
3
请输入要求的两数之和B：
1
请输入3个正整数的序列（空格隔开）：
1 2 3
第2组测试数据
3个不同的正整数： 1 2 3
正整数B： 1
查找结果： 有0对整数对：

第3组测试数据
3个不同的正整数： 1 2 3
正整数B： 4
查找结果： 有1对整数对： 1 3

```

```
请输入要处理的正整数个数（输入-1以退出）：
3
请输入要求的两数之和B：
4
请输入3个正整数的序列（空格隔开）：
1 2 3
第3组测试数据
3个不同的正整数：1 2 3
正整数B：4
查找结果：有1对整数对：1 3

请输入要处理的正整数个数（输入-1以退出）：
-1
请按任意键继续...
```

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）。

