期中-上机部分

1

5.1 请编写 C 语言函数，实现一个部分有序数组（分为 2 个有序子序列）的排序。

例如：数组 int a[10] = { 1，3，5，7，9，10，2，4，6，8 }分为两个子序列：a[left]至 a[mid]； a[mid+1]至 a[right]分别有序，其中 left = 0，mid = 5，right = 9。

要求：使用临时数组 int temp[ ] （与数组 a 大小相同），利用数组部分有序的特点实现数组 a 的排序。

函数头定义如下：

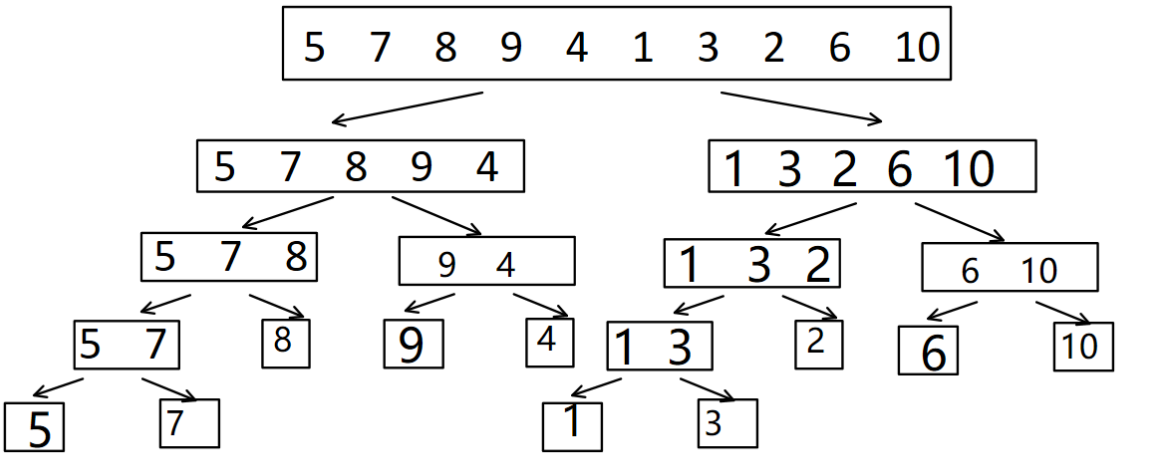
int merge(int a[], int temp[], int left, int mid, int right)

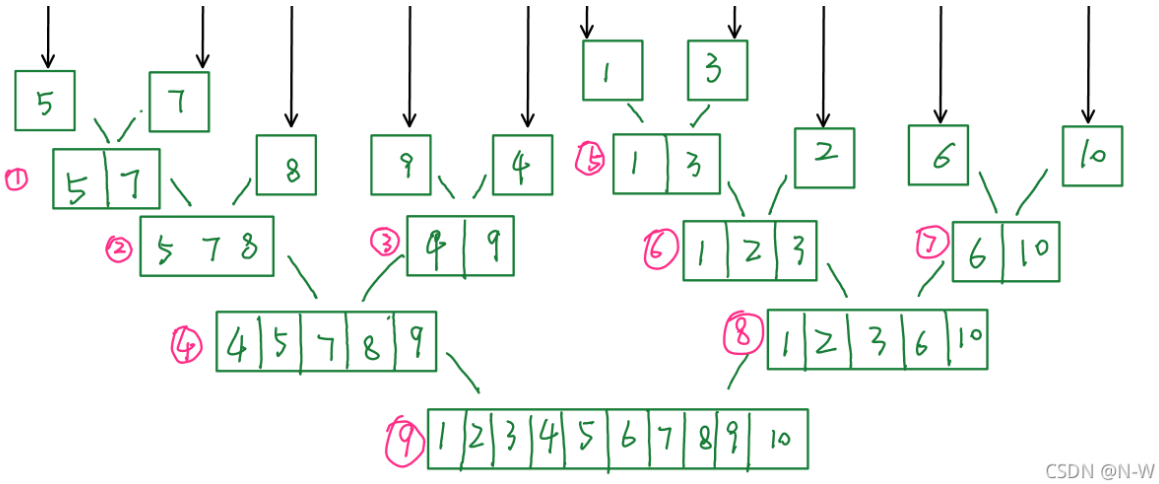
```c
#include<stdio.h>
#define N 10

void show(int a[], int left, int right) {
    int i;
    for (i = left; i <= right; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");
}

int merge(int a[], int temp[], int left, int mid, int right) {
    int i1 = left, i2 = mid + 1, i = 0;
    while (i1 != mid + 1 || i2 != right + 1) {
        if (i1 == mid + 1)
            temp[i++] = a[i2++];
        else if (i2 == right + 1)
            temp[i++] = a[i1++];
        else if (a[i1] <= a[i2])
            temp[i++] = a[i1++];
        else
            temp[i++] = a[i2++];
    }
    for(i =0; i <= right - left; i++) {
        a[left + i] = temp[i];
    }
    return 0;
}


int main() {
    int a[N] = { 1,3,5,7,9,10,2,4,6,8 }, temp[N];
    int left, mid, right;
    left = 0, mid = 5, right = 9;
    show(a, left, right);
```

```
36      merge(a, temp, left, mid, right);
37      show(a, left, right);
38  }
```

1. 按顺序比较两个有序子序列的元素大小，temp记录较小的值并将对应子序列的索引加一
2. 插入排序（将后半部分序列元素逐个插入到前半部分序列中，同样利用了部分有序的特点，但插入时数组中间部分要整体后移一位，移位循环条件容易出错且复杂度较高）
3. *直接冒泡排序（不推荐，没有利用部分有序的特点，时间复杂度由O(N)上升为$O(N^2)$）*

- 实际应用：merge为归并排序的其中一个步骤 https://blog.csdn.net/weixin_50941083/article/details/120852477





问题

1. 未能完全理解题意，仅实现了将数组划分为两部分分排序的功能；**部分有序是前提而非目的**
2. **未考虑边界情形**，当其中一个子序列遍历结束后，未对另一子序列中剩余元素进行处理
3. 冒泡排序和选择排序模板混淆，两者在内循环的遍历方向不同；插入排序内循环条件不应该与固定值比较，因为随着元素不断插入，已排序序列的长度也随之增加

2

5.2 编写实现以下要求的程序，键盘输入字符串 s，找到 s 中最长的回文子串。

示例 输入：s= cabccbad 输出：abccba

```
```

```c
#define  _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<malloc.h>
#define N 100

char* longestPalindromicSubstring(char *s) {
    int n = strlen(s);
    int i, j;
    int length = 0;
    //char *string = malloc(sizeof(char) * N);
    static char string[N] = "";
    for (i = 0; i < n; i++) {
        for (j = 1; i - j >= 0 && i + j < n; j++) {
            if (s[i - j] != s[i + j]) break;
        }
        j--;
        if (length < j * 2 + 1) {
            length = j * 2 + 1;
            strncpy(string, s + i - j, length);
            string[length] = 0;
        }
    }
    for (i = 0; i < n; i++) {
        for (j = 0; i - j >= 0 && i + j + 1 < n; j++) {
            if (s[i - j] != s[i + j + 1]) break;
        }
        j--;
        if (length < j * 2 + 2) {
            length = j * 2 + 2;
            strncpy(string, s + i - j, length);
            string[length] = 0;
        }
    }
    return string;
}


int main() {
    char s[N] = "cabccbad";
    //gets(s);
    printf("%s\n", longestPalindromicSubstring(s));
}
```

```c
#define  _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<malloc.h>
#define N 100

char* longestPalindromicSubstring2(char *s) {
    int n = strlen(s);
    int i, j;
    int length;
    static char string[N] = "";
```

```
    for (length = n; length > 0; length--) {
        for (i = 0; i + length <= n; i++) {
            for (j = 0; j < (length >> 1); j++) {
                if (s[i + j] != s[i + length - 1 - j])
                    break;
            }
            if (j == (length >> 1)) {
                strncpy(string, s + i, length);
                string[length] = 0;
                return string;
            }
        }
    }
    return string;
}


int main() {
    char s[N] = "cabccbad";
    //gets(s);
    printf("%s\n", longestPalindromicSubstring2(s));
}
```

1. 遍历字符串中的每个字符，分别考虑以该字符或该字符与下一字符为中心所能构成的最大回文子串
2. 从大到小枚举所有可能的回文子串长度，检查相应长度的子串是否回文

- 题目没有要求子函数实现，当在main函数中时原局部变量string可以不考虑堆空间申请或标记为static存储于全局区（防止函数返回时清除栈上变量），可直接申请栈空间。

问题

1. 当检查回文字符串的方向是从中间向两端时，特别注意要分别考虑**奇数长度和偶数长度**的回文字符串
2. 仅判断输入字符串是否回文，或者列举出所有回文子串，并未保存或输出最大回文子串

其他问题

程序运行截图

1. 使用windows自带截图工具：按下`win`，输入“截图工具”，选择菜单栏-“新建”或直接按下`ctrl`+`n`
2. 按下`PrtSc`，在word中新建空白文档，按下`ctrl`+`v`，然后修改桌面截图
3. 使用其他工具软件，如微信、QQ、FastStone Capture等
4. 拍照（会有摩尔纹和反光）

- 以后未见程序运行截图时，按未完成处理或额外扣分

注释

- 合适的注释能够体现想法与思路，在一定程度上也能提高分数，相当于伪代码

输入输出

- 初步调试程序时，不必使用输入输出，可以在程序内先使用实例给出的固定参数并通过局部变量窗口或监视窗口来调试代码，可大大节省调试时额外的输入时间。

C++

- 源代码文件后缀名应为.c而非.cpp
- 不要使用`#include<iostream>` `#include<bits/stdc++.h>` `cout` `cin` `using namespace std;` 等C++语法，此内容不在本课程C语言的范畴内
    - 22307140108