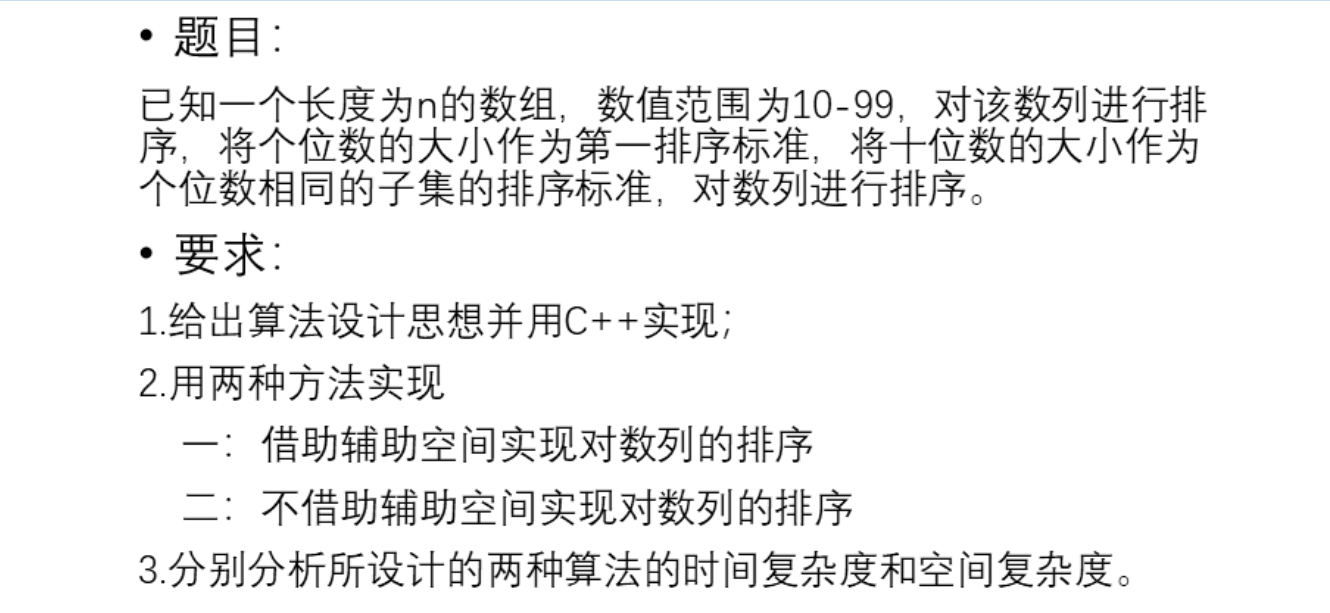
数据结构实验一

班级：20信安法 学号：2013484 姓名：张世伟

1. 实验内容



1. 设计思路
2. 不借助辅助空间

首先将数组中的所有数字个位数与十位数互换位置，再对换位后的数进行排序，最后将排序后的数个位数和十位数互换位置。

1. 借助辅助空间

首先根据个位数的大小进行排序，并借助辅助数组记录不同个位数的数量，最后根据辅助数组记录的数据和十位数的大小进行排序。

1. 核心代码
2. 不借助辅助空间

//不借辅助空间2

//一、将个位和十位调换位置

//二、排序

//三、将个位和十位位置换回来

void \_swap(int\* ptr1)

{

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

{

if (ptr1[i] > 9)

ptr1[i] = ptr1[i] / 10 + ptr1[i] % 10 \* 10;//使用这个方式调换位置会出现70换成7状况，所以分类

else

ptr1[i] \*= 10;

}

}

//快排

int solution(int\* nums, int left, int right)

{

int pivot = nums[right];

int i = left - 1;

for (int j = left; j < right; j++)

{

if (nums[j] <= pivot)

{

i += 1;

swap(nums[i], nums[j]);

}

}

swap(nums[i + 1], nums[right]);

return i + 1;

}

void quicksort(int\* nums, int left, int right)

{

if (left < right)

{

int pos = solution(nums, left, right);

quicksort(nums, left, pos - 1);

quicksort(nums, pos + 1, right);

}

}

1. 借助辅助空间

int temp[10];

//借助辅助空间1

//先根据个位进行排序

//再根据十位进行排序

void compositor(int\* ptr)

{

//使用冒泡排序

//根据个位进行第一轮排序

//借助辅助空间计个位数相同的数的数量

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

temp[ptr[i] % 10]++;

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

{

for (int j = i + 1; j < n; j++)

{

if ((ptr[i] % 10) > (ptr[j] % 10))

{

swap(ptr[i], ptr[j]);

}

}

}

//根据辅助数组所计数量和十位数进行第二轮排序

int sum = 0;//用来表示需要进行排序的第一个数字的位置

for (int k = 0; k < 10; k++)

{

for (int i = sum; i < temp[k]+sum; i++)

{

for (int j = i + 1; j < temp[k]+sum; j++)

{

if ((ptr[i] / 10) > (ptr[j] / 10))

{

swap(ptr[i], ptr[j]);

}

}

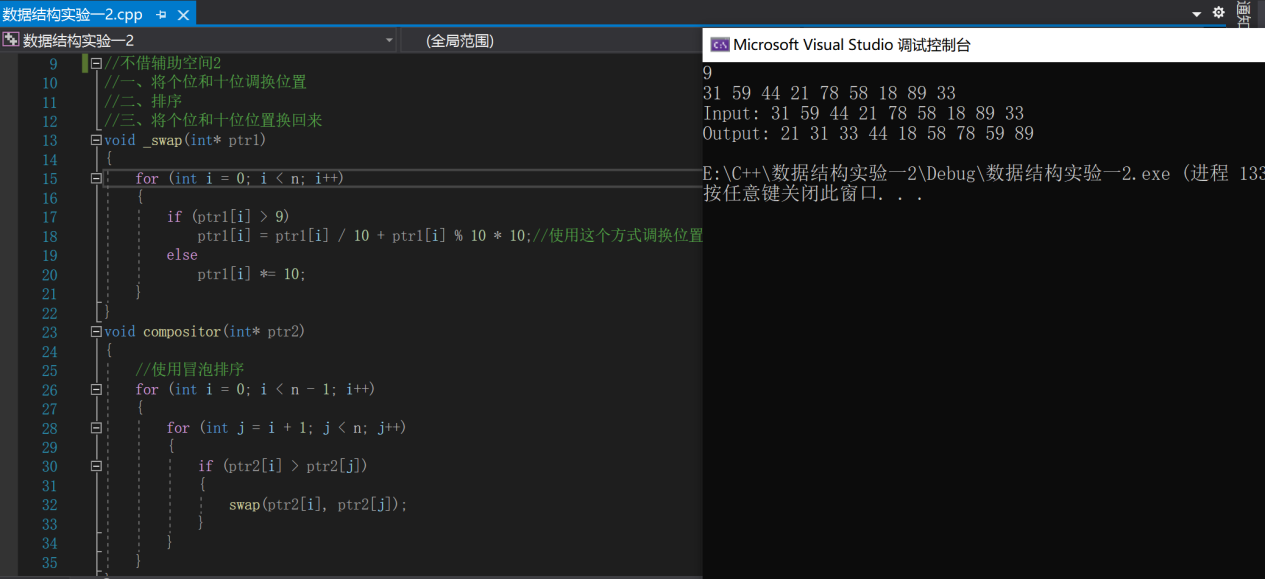
}

sum += temp[k];

}

}

1. 运行结果





1. 实验分析
2. 不借助辅助空间
3. 空间复杂度
4. 时间复杂度
5. 借助辅助空间
6. 空间复杂度
7. 时间复杂度

借助辅助空间与不借助辅助空间所需要的空间复杂度相同，所需要的时间复杂度主要取决于排序使用的算法