

《数据结构》实验报告

班 级: _____

姓 名: _____

学 号: _____

E-mail: _____

日 期: _____

◎实验题目:

1.1

Description

给定两个按照升序排列的有序数组，请把它们合成一个升序数组并输出。

1.2

实验1.2: 高精度计算PI值

Time Limit: 3000ms , Memory Limit: 10000KB , Accepted: 0 , Total Submissions: 0

Description

限制使用双向链表作存储结构，请根据用户输入的一个整数（该整数表示精确到小数点后的位数，可能要求精确到小数位），高精度计算PI值。可以利用反三角函数幂级展开式来进行计算。

◎实验内容:

1.1

Input

第一行为第一个有序数组的长度，正整数 n ， $n \leq 20$ ；
第二行为第一个有序数组的 n 个数字，用空格隔开；
第三行为第二个有序数组的长度，正整数 m ， $m \leq 20$ ；
第四行为第二个有序数组的 m 个数字，用空格隔开。

Output

输出合并后的数组，每个数字占一行。

1.2

Input

第一行为第一个有序数组的长度，正整数 n ， $n \leq 20$ ；
第二行为第一个有序数组的 n 个数字，用空格隔开；
第三行为第二个有序数组的长度，正整数 m ， $m \leq 20$ ；
第四行为第二个有序数组的 m 个数字，用空格隔开。

Output

输出合并后的数组，每个数字占一行。

一、需求分析

1.1

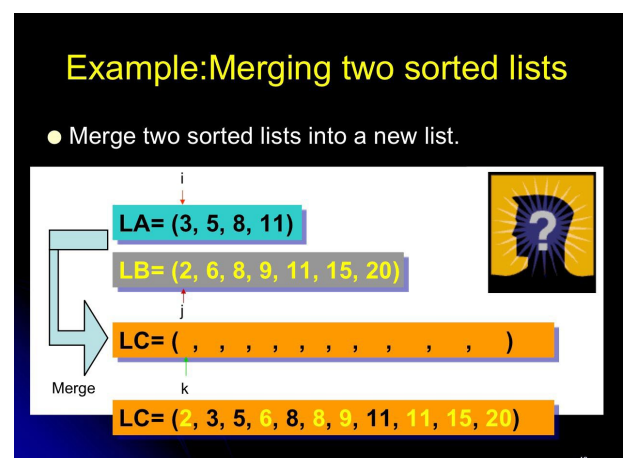
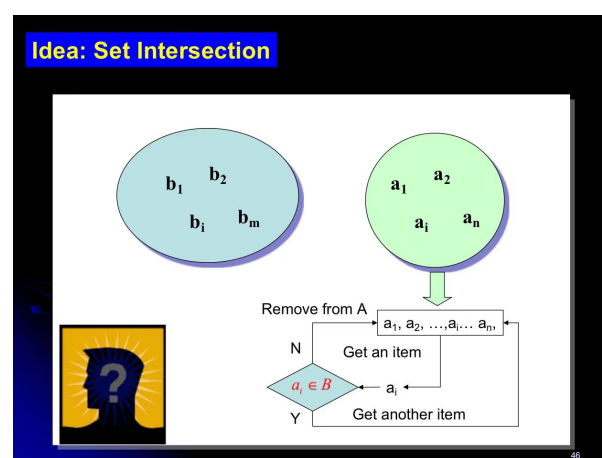
本实验输入两个有序数组，输出一个合并后的广义升序数组，核心就是要依次比较两个数组中各元素的大小，并将其存入新数组中；

1.2

本实验要求高精度的去计算 PI，并输出精确到 n 位的值，结合数学知识可知，运用反三角函数和泰勒公式可以近似求出符合要求的 PI 值。

二 概要设计

1.1



1.2

有公式: $\frac{\pi}{2} = \sum_{n=0}^{\infty} \frac{n!}{(2n+1)!!}$

即 $R(n+1) = R(n) * \frac{n}{2n+1}$, $R(1) = 1$, $\text{sum} = \pi = 2 * \sum_{n=1}^{\infty} R(n)$

<https://blog.csdn.net/abcdef0623>

三 详细设计

1.1

对于已经是广义升序的输入数组，且元素均为整形，主要思路是依次从两个数组中各取出一个元素进行比较，小的先放入新数组，大的后放入，在这个过程中采用一个变量记录待填入数据的位置。

数据结构: SeqList 顺序表，内含 elenum（元素个数）和 list[] 两个元素。

算法设计:

CreateSeq: 构建顺序表，将 pList 和元素个数 n 作为输入值，采用循环输入填充数组。

MergeList: 按照思路，不断比较两个数组的元素并将其填入新数组，当一个数组为空时，将未遍历完的数组的剩余元素直接填入新数组。

PrintList: 在遍历完成后，将新数组的结构体指针作为传入值，将得到的新数组采用循环输出，每次输出均换行的形式打印出来。

1.2

对于要高精度计算 PI 的值，结合已有的数学知识，主要核心是依赖采用的数据结构进行大数的四则运算，根据所要求的精度重复计算，最终得到高精度值。

数据结构：DList 双向链表，依赖于 DNode 双向节点（包括 data, pNode next, pNode pre 三部分）进行构建。

算法设计：

pNode Init()：初始化双向链表为空，为表头分配空间；

pNode CreateDList(pNode head)：创建双向链表，采用循环为新节点分配空间并初始化所有节点 data 为 0；

主函数：

1. 接受输入的精度 n；
2. 初始化双向链表指针 num(存放每一次计算出的 $R(n)$ 值)，sum(存放所有 $R(n)$ 之和)；
3. 后续计算思路：利用递推公式 $R(n+1) = R(n) * n / (2*n+1)$ 计算 $R(n)$ ，将其存入 num 中，先算乘法再算除法，然后累加，结果存到 sum 中；
4. 数据处理：先计算乘法 $R(n) * n$ ，利用大数乘法的思路，从后往前算，模拟手算进位，每个节点存放一位数，用 ret 存放进位后剩余的数；
5. 数据处理：计算 $R(n) * n / (2*n+1)$ ，利用大数除法的思路，模拟手算，前一位需要 *10，每个节点存放一位数，用 ret 存放余数；
6. 数据处理：使用大数加法，从后往前算，模拟手算进位，将相加后得到的数据存储在 sum 中；
7. 先输出 "3."，然后按照 n 位的精度，循环输出 sum 里的值。

四 使用说明、测试分析及结果

1、说明如何使用你编写的程序；

1.1:

输入第一个数组的元素个数，然后依次输入元素值；

输入第二个数组的元素个数，依次输入元素值；

输出合并后的数组；

1.2:

输入数字 n, 输出 π 的近似值，结果保留小数点后 n 位；

2、测试结果与分析；

1.1:

输入：

数组 1 长度：3，元素：1 3 7；

数组 2 长度：5，元素：2 4 6 8 10；

输出：

每行一个数字：1 2 3 4 6 7 8 10.

1.2:

输入：9

输出：3.141592653

3、调试过程中遇到的问题是如何解决以及对设计与实现的回顾讨论和分析

1.1:

使用顺序表时，注意构建数组时要先规定好数组最大长度，避免数组越界，同时遍历完一个数组后要直接填入另一数组，节省时间，还需要注意判断数组遍历结束的条件；

1.2:

①在考虑计算 PI 值时，先要确定好采用的数学方法，即使用反三角函数和泰勒展开计算 PI 值；

②在思考如何计算时：一开始只想到了单向链表，但在构建过程中发现，单线链表只能满足一个方向的大数计算，而乘法和加法为从后往前算，除法为从前往后算，因此将单向链表改为双向链表，得以成功进行数值计算；

③在储存数据时，开始没有想到用 temp 储存临时数据，后续及时意识到；

4、运行界面

1.1:

```
(base) PS D:\CodespaceX\Cpp\data st
++ nojT1.cpp -o nojT1 } ; if ($?) {
3
1 3 7
5
2 4 6 8 10
1
2
3
4
6
7
8
10
```

1.2:

```
(base) PS D:\CodespaceX\
++ nojT2.cpp -o nojT2 }
9
3.141592653
```

五、实验总结

①第一题因为在之前已做过类似的题，因此花费时间较短，第二题由于开始不清楚具体的 PI 值估算方法，自己摸索了很久，最后查询到了相应的数值计算方法，未来可以加强数学方面的学习；

②第二题的除法大数运算时，没有及时想到要重置存储余数的 ret=0，而是沿用了乘法运算后的 ret，导致开始调试错误，以后对于在不同循环中要重复使用的功能性变量要及时重置；

③应该更多采用函数式编程，但最后因为时间原因，第二题没有将主要的数值计算封装为函数，后续可以改进。

教师评语：

实验成绩：

指导教师签名：

批阅日期：