《数据结构》课程实践报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院系 | 计算机学院 | | 年级专业 | 23软工 | 姓名 | 王天予 | 学号 | 2362401031 |
| 实验布置日期 | | 2024.9.24 | | 提交  日期 |  | | 成绩 |  |

## 课程实践实验2-2：大整数的实现与运算

### 一、问题描述及要求

在计算机科学中，传统的整型数据类型有其限制，无法处理非常大的整数，例如超过64位的数字。为了解决这个问题，我们需要实现一个大整数类，支持对大整数的基本运算，包括加法、减法、乘法和除法。大整数将以字符串形式存储，以便于处理任意大小的整数。要求编写程序，支持大整数的输入、存储以及基本运算。

### 二、概要设计

1. 实验内容理解  
   大整数类的实现涉及字符串处理、动态内存管理和基本算法的应用。我们的目标是设计一个灵活的类，能够高效地处理大整数的各种运算。
2. 系统功能列表  
   1）大整数输入：从用户输入或文件读取大整数。  
   2）加法运算：实现两个大整数的相加。  
   3）减法运算：实现两个大整数的相减。  
   4）乘法运算：实现两个大整数的相乘。
3. 程序运行界面设计  
   提供命令行交互式界面，用户可以输入指令来执行各种操作，如add（加法）、subtract（减法）、multiply（乘法）和quit（退出程序）。
4. 总体设计思路  
   使用C++中的字符串类来存储大整数，所有的运算均通过对字符串的逐位处理实现。核心类为BigInteger，其中包含了大整数的初始化、运算和输出等功能方法。
5. 程序结构设计  
   主程序文件（BigInteger.cpp）：包含BigInteger类的定义和实现，以及主函数main和命令行交互逻辑。  
   类设计：
   * BigInteger类：负责大整数的存储和运算。
     + 主要方法：  
       1）add(const BigInteger &other)：执行加法运算。  
       2）subtract(const BigInteger &other)：执行减法运算。  
       3）multiply(const BigInteger &other)：执行乘法运算。
6. 算法和数据结构选择理由  
   使用字符串存储大整数，可以灵活处理任意大小的数字，避免了整型溢出的问题。通过逐位处理运算，保证了算法的准确性和可扩展性。

### 三、详细设计

1. 主函数设计  
   main函数中首先读取用户输入的操作类型，然后创建BigInteger对象实例，并调用相应的运算方法处理用户输入。
2. 关键算法设计  
   1）大整数加法算法：从低位到高位逐位相加，考虑进位情况。  
   2）大整数减法算法：从低位到高位逐位相减，考虑借位情况。  
   3）大整数乘法算法：使用逐位乘法和加法的结合，处理每一位的乘法和进位。
3. 程序优化  
   在实现过程中，避免了对整个大整数的重复遍历，利用字符串操作和辅助函数，提高了运算效率。

### 四、实验结果

1. 测试用例 1:

* 测试输入序列：

add 12345678901234567890 98765432109876543210

* 测试目的：  
  测试程序的加法功能，验证结果的准确性。
* 正确输出：

111111111011111111100

* 实际输出：

111111111011111111100

* 错误原因：  
  无。
* 测试结论：  
  通过。程序正确实现了大整数加法。

1. 测试用例 2:

* 测试输入序列：

subtract 10000000000000000000 9999999999999999999

* 测试目的：  
  验证程序的减法功能。
* 正确输出：

1

* 实际输出：

1

* 错误原因：  
  无。
* 测试结论：  
  通过。程序正确实现了大整数减法。

1. 测试用例 3:

* 测试输入序列：

multiply 123456789 987654321

* 测试目的：  
  测试乘法运算的实现。
* 正确输出：

121932631112635269

* 实际输出：

121932631112635269

* 错误原因：  
  无。
* 测试结论：  
  通过。乘法功能正常。

### 五、实验分析与探讨

1. 测试结果分析  
   所有基础功能测试均通过，程序能够根据输入的指令正确执行大整数的运算。通过手动输入和预定义测试用例，验证了大整数类的核心功能。
2. 性能分析
   * 时间复杂度：大整数的加法和减法算法时间复杂度为O(n)，乘法为O(n^2)
   * 空间复杂度：主要受存储大整数字符串的影响，空间复杂度为O(n)，n为大整数的位数。

### 六、小结

本次实验通过实现大整数类，加深了对字符串处理、动态内存管理及算法实现的理解。未来可以扩展功能，如支持大整数的比较、模运算等。此次实践不仅提升了编程能力，也为大数处理的相关应用提供了良好的基础。