长整数加法思路提示

主讲人 周奕端

纲要

- 第一部分：流程分析
- 第二部分：数据结构设计
- 第三部分：算法实现

第一部分：流程分析

- 从文件获取长整数字符串；
- 从字符串获取两个M进制数；
- 判断调用算法，调用大整数加法还是大整数减法（输入数据的4种情况，正+正，正+负，负+正，负+负，均可变为正数+正数和大正数-小正数）；
- 调用算法；
- 进制转换为N。

纲要

- 第一部分：流程分析
- **第二部分：数据结构设计**
- 第三部分：算法实现

第二部分：数据结构设计

- 长整数可以采用符号+绝对值的方式存储
- 绝对值：vector<int>
- 符号：int
- 数据结构：

```
struct LongInt
{
    std::vector<int> data;
    int signal;
};
```

纲要

深蓝学院
shenlanxueyuan.com

- 第一部分：流程分析
- 第二部分：数据结构设计
- **第三部分：算法实现**

第三部分：算法实现

深蓝学院
shenlanxueyuan.com

- Step1：从文件获取长整数字符串
  - 存在多种方法，使用ifstream与getline配合比较简单。

```
std::ifstream file("name");
if(file.open())
{
    std::string str;

    while(std::getline(file, str))
    {
        // TODO:
    }
}
```

第三部分：算法实现

- Step2：从字符串转换得到两个M进制LongInt
  - 判断str[0] == ‘+’ / str[0] == ‘-’，并将长整数signal设为对应数值；
  - 循环移除前导零，可以只记录下标而不去真的删除零；
  - 数字字符可以用ch – ‘0’来求事实上的值，字符可以根据大小写，分别用ch – ‘A’和ch – ‘a’来计算值；
  - 由于加法和减法一般从低位计算，所以在LongInt中存放时可以考虑倒序存放。

第三部分：算法实现

- Step3：判断调用函数
  - 若为两正数，调用相加函数；若为两负数，则转换为负的两正数之和，调用相加函数；
  - 若为一正一负相加，则比较绝对值，变为绝对值较大减绝对值较小，调用相减函数；
  - 绝对值比较：先比较位数，位数多者更大；若位数相同，则从高位依次比较LongInt::data，直到比较出大小。

第三部分：算法实现

深蓝学院 shenlanxueyuan.com

- Step3.1：长正整数加法实现
  - 考虑第k位，设sum为数字a和b第k位之和再加上第k-1位进位的数值，则结果的第k位为x%M，同时第k位向第k+1位进位x/M；
  - 近似代码为：

```
int x = a.data[k] + b.data[k] + carry[k - 1];
res.data[k] = x % M;
carry[k] = x / M;
```

第三部分：算法实现

深蓝学院 shenlanxueyuan.com

- Step3.2：长正整数减法实现
  - 考虑第k位，设sub为数字a减去第k-1位借位的数值再和b第k位之差，若有借位，则结果的第k位为x+M，否则第k位为x；
  - 近似代码为：

```
int x = a.data[k] - borrow[k - 1] - b.data[k];
res.data[k] = (x < 0) ? (x + M) : x;
borrow[k] = (x < 0) ? 1 : 0;
```

第三部分：算法实现

深蓝学院 shenlanxueyuan.com

- Step4：进制转换
  - M进制转N进制，可以用短除法的方式求解；
  - 如十进制数12321496转八进制可以用如下步骤：
    - 12321496除以8等于1540187余0
    - 1540187除以8等于192523余3
    - 192523除以8等于24065余3
    - 24065除以8等于3008余1
    - 3008除以8等于376余0
    - 376除以8等于47余0
    - 47除以8等于5余7
    - 5除以8等于0余5
  - 最终结果为余数倒序，即$(57001330)_8$。

第三部分：算法实现

- Step4：进制转换（续1）
  - 我们需要实现每一次短除法，这里采用模拟除法的方式；
  - 第k位除以进制N时，其得到的那一位的商即最终商的对应位，而余数用于与k+1位共同构成求解商下一位的值；

除以8	1	2	3	2	1	4	9	6
商	0	1	5	4	0	1	8	7
余	1	4	3	0	1	6	5	0

  - 每一次的商从左到右为短除法下一次的被除数，余数的最后一位是答案的对应位。

第三部分：算法实现

深蓝学院 shenlanxueyuan.com

- Step4：进制转换（续2）
  - 以上算法的近似代码为：

```
x = remainder.data[k - 1] * M + number.data[k];
quotient.data[k] = x / N;
remainder.data[k] = x % N;
```

  - 得到则quotient去除前导零之后即为下一个被除数，remainder的最后一位即为答案的长整数中的一位。

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening