## 算法题-大数相乘问题

今天在网上看到一个大数相乘的问题，题目是这样的：输入两个整数，要求输出这两个数的乘积。输入的数字可能超过计算机内整形数据的存储范围。

分析：

由于数字无法用一个整形变量存储，很自然的想到用字符串来表示一串数字。然后按照乘法的运算规则，用一个乘数的每一位乘以另一个乘数，然后将所有中间结果按正确位置相加得到最终结果。可以分析得出如果乘数为A和B，A的位数为m，B的位数为n，则乘积结果为m+n-1位（最高位无进位）或m+n位（最高位有进位）。因此可以分配一个m+n的辅存来存储最终结果。为了节约空间，所有的中间结果直接在m+n的辅存上进行累加。最后为了更符合我们的乘法运算逻辑，可以讲数字逆序存储，这样数字的低位就在数组的低下标位置，进行累加时确定下标位置较容易些。

下面是我的解法。

首先是对数组逆序的函数：

```
1 void reverseOrder(char* str, int p, int q)
2 {
3     char temp;
4     while(p < q)
5     {
6         temp = str[p];
7         str[p] = str[q];
8         str[q] = temp;
9         p ++;
10        q --;
11    }
12 }
```

然后是完成大数相乘的函数：

```
1 char* multiLargeNum(char* A, char* B)
2 {
3     int m = strlen(A);
4     int n = strlen(B);
5     char* result = new char[m+n+1];
6     memset(result, '0', m+n);
7     result[m+n] = '\0';
8     reverseOrder(A, 0, m-1);
9     reverseOrder(B, 0, n-1);
10
```

文章分类

.NET
C#
C/C++
Linux(1)
Python
Qt
Ubuntu(1)
算法

阅读排行榜

1. 算法题-大数相乘问题(2245)
2. 算法题-字符串循环移位问题(1613)
3. Python小应用1 - 抓取网页中的链接地址(1106)
4. const修饰指针的情况分析(299)
5. 微软面试记录(165)

评论排行榜

1. const修饰指针的情况分析(4)
2. 算法题-字符串循环移位问题(2)

推荐排行榜

1. 算法题-字符串循环移位问题(4)

```
11      int multiFlag; // 乘积进位
12      int addFlag;   // 加法进位
13      for(int i=0; i <= n-1; i++) // B的每一位
14      {
15          multiFlag = 0;
16          addFlag = 0;
17          for(int j=0; j <= m-1; j++) // A的每一位
18          {
19              // '0' - 48 = 0
20              int temp1 = (A[j] - 48) * (B[i] - 48) + multiFlag;
21              multiFlag = temp1 / 10;
22              temp1 = temp1 % 10;
23              int temp2 = (result[i+j] - 48) + temp1 + addFlag;
24              addFlag = temp2 / 10;
25              result[i+j] = temp2 % 10 + 48;
26          }
27          result[i + m] += multiFlag + addFlag;
28      }
29      reverseOrder(result, 0, m+n-1); // 逆序回来
30
31      return result;
32  }
```

最后是测试程序：

```
1  int main()
2  {
3      char A[] = "962346239843253528686293234124";
4      char B[] = "93459382645998213649236498";
5      char *res = multiLargeNum(A, B);
6      if(res[0] != 48)
7          printf("%c", res[0]);
8      printf("%s", res+1);
9      delete [] res;
10     return 0;
11 }
```

时间复杂度分析：

3个逆序操作的时间分别为O(n)、O(m)、O(m+n)，双重循环的时间复杂度为O(mn)，则总的时间复杂度为O(mn + (m+n))，通常m+n << mn，因此可近似认为为O(mn)。而且，逆序操作只是为了思考更容易，完全可以去掉。

空间复杂度为O(m+n)

分类: C/C++, 算法



Jason Yang
关注 - 0
粉丝 - 5
+加关注



« 上一篇：算法题-字符串循环移位问题

posted on 2012-04-26 22:56 Jason Yang 阅读(2246) 评论(0) 编辑 收藏