

原创

模拟法计算大数加减乘除 (附带例题)

2019-07-16 21:11:20

Y-Y_

阅读数 560

文章标签:

大数加法

大数减法

大数除法

大数乘法

更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/weixin_44410512/article/details/96098341

模拟法计算大数加减乘除就是模拟列竖式的方法计算

大数加法

模拟竖式加法

首先是第一步先最低位对齐吧

```
1 | 例如：
2 | 1 2 3 4 5 6 7 8 9
3 |   2 0 0 0 1 1 2 8
4 | 我是这样子在实现的
5 | strrev(a)//char的反转函数
6 | reverse(a.begin(),a.end());//string的反转函数
7 | 反转过来就是这个
8 | 9 8 7 6 5 4 3 2 1
9 | 8 2 1 1 0 0 0 2
```

第二步就是上下相加，满10进位

```
1 | 例如：
2 | 1 2 3 4 5 6 7 8 9
3 |   2 0 0 0 1 1 2 8
4 | -----
5 |           1 1
6 | 1 4 3 4 5 7 9 1 7
7 | 因为我已经反转过来了，那就是这样的
8 | 9 8 7 6 5 4 3 2 1
9 | 8 2 1 1 0 0 0 2
10 | -----
11 |    1 1
12 | 7 1 9 7 5 4 3 4 1
13 | 这样看起来很好加吧，一个for循环跑到天荒地老，（当然了，要是没得加就得退出了）
14 | 然后在反转一下就可以了
15 | 1 4 3 4 5 7 9 1 7
```

很简单吧，这样就计算出A+B了

例题

<http://sustoj.com/JudgeOnline/problem.php?id=1952>

AC code:

```
1 | #include <stdio.h>//c 版
2 | #include <string.h>
3 | #define maxn 10000500
4 | char ans[maxn];
5 | char *strrev(char *str){
6 |     char *p1, *p2;
7 |     if (! str || ! *str){
8 |         return str;
9 |     }
10 |     for (p1 = str, p2 = str + strlen(str) - 1; p2 > p1; ++p1, --p2){
11 |         *p1 ^= *p2;
12 |         *p2 ^= *p1;
13 |         *p1 ^= *p2;
14 |     }
15 |     return str;
```

编程语言大PK，你选谁？

关闭

```

16 }
17 const char*num_plus(char *a,char *b){
18     memset(ans,0,sizeof ans);
19     int lena=strlen(a);
20     int lenb=strlen(b);
21     strrev(a);
22     strrev(b);
23     int yu=0;
24     for(int i=0;;i++){
25         if(i<lena&&i<lenb){
26             ans[i]=(yu+a[i]-'0'+b[i]-'0')%10+'0';
27             yu=(yu+a[i]-'0'+b[i]-'0')/10;
28         }else if(i<lena){
29             ans[i]=(yu+a[i]-'0')%10+'0';
30             yu=(yu+a[i]-'0')/10;
31         }else if(i<lenb){
32             ans[i]=(yu+b[i]-'0')%10+'0';
33             yu=(yu+b[i]-'0')/10;
34         }else if(yu!=0){
35             ans[i]=yu%10+'0';
36             yu=yu/10;
37         }else break;
38     }
39     strrev(ans);
40     return ans;
41 }
42 char a[maxn],b[maxn];
43 int main(){
44     while(~scanf("%s %s", a, b)){
45         printf("%s\n", num_plus(a,b));
46     }
47     return 0;
48 }
49
50

```

```

1 #include <cstring>//c++ 版
2 #include <iostream>
3 #include <algorithm>
4 using namespace std;
5 string ans;
6 string num_push(string a,string b){
7     ans.clear();
8     int lena=a.size();
9     int lenb=b.size();
10    reverse(a.begin(),a.end());
11    reverse(b.begin(),b.end());
12    int yu=0;
13    for(int i=0;;i++){
14        if(i<lena&&i<lenb){
15            ans+=(yu+a[i]-'0'+b[i]-'0')%10+'0';
16            yu=(yu+a[i]-'0'+b[i]-'0')/10;
17        }else if(i<lena){
18            ans+=(yu+a[i]-'0')%10+'0';
19            yu=(yu+a[i]-'0')/10;
20        }else if(i<lenb){
21            ans+=(yu+b[i]-'0')%10+'0';
22            yu=(yu+b[i]-'0')/10;
23        }else if(yu!=0){
24            ans+=yu%10+'0';
25            yu=yu/10;
26        }else break;
27    }
28    reverse(ans.begin(),ans.end());
29    return ans;
30 }
31 string a,b;
32 int main(){
33     while(cin>>a>>b){
34         cout<<num_push(a,b)<<endl;

```

```
35 | }
36 |     return 0;
37 | }
38 |
39 |
```

大数减法

模拟竖式减法

首先是第一步要判读大小吧（因为我只学过大减小的竖式计算）

```
1 | 因为没有前置零且都是正整数
2 | 所以谁位数多谁就大
3 | 要是位数一样的话，
4 | 那就是谁的那字典序大谁就大
5 | 用到了 strcmp(a,b) //char 类型的比较函数
```

字典序定义：

简单理解 [\[编辑 \]](#)

设想一本英语字典里的单词，哪个在前哪个在后？

显然的做法是先按照第一个字母、以 a、b、c.....z 的顺序排列；如果第一个字母一样，那么比较第二个、第三个乃至后面的字母。如果比到最后两个单词不一样，sigh 和 sight），那么把短者排在前。

第二步就是最低位对齐

```
1 | 例如：
2 | 1 2 3 4 5 6 7 8 9
3 |   2 0 0 0 1 1 2 8
4 | 我是这样子在实现的
5 | strrev(a)//char的反转函数
6 | reverse(a.begin(),a.end());//string的反转函数
7 | 反转过来就是这个
8 | 9 8 7 6 5 4 3 2 1
9 | 8 2 1 1 0 0 0 2
```

第三步就是上下相减了，借位

```
1 | 例如：
2 | 1 2 3 4 5 6 7 8 9
3 |   2 0 0 0 1 1 2 8
4 | -----
5 | 1 0 3 4 5 5 6 6 1
6 | 因为我已经反转过来了，那就是这样的
7 | 9 8 7 6 5 4 3 2 1
8 | 8 2 1 1 0 0 0 2
9 | -----
10 | 1 6 6 5 5 4 3 0 1
11 | 这样看起来很好减吧，一个for循环跑到天荒地老，（当然了，要是没得减就得退出了）
12 | 然后在反转一下就可以了
13 | 1 0 3 4 5 5 6 6 1
```

借位

```
1 | 1 0 0 0 0
2 |   9 9 9 9
3 | -----
4 | 0 0 0 0 1
5 | 我是这样实现借位的，先反转过来
6 | 0 0 0 0 1
7 | 9 9 9 9
8 | -----
9 | 因为0减不过9
10 | 所以要向后借位
11 | 所以第一位借到10后就变成了1
12 | 第二位是'0'的话就继续向后借，
```

编程语言大PK，你选谁？

关闭

```

13 反正上面的肯定大于下面的，所以肯定能借到的
14 借到就减一就行了
15 就是这个样子了
16
17 0 0 0 0 1
18 9 9 9 9
19 -----
20 1 0 0 0 0
21 翻转过来就是
22 0 0 0 0 1 了
23 当然不能要前置0，所以我在反转前就把前置零去掉，不要忘了要是小减小就要添上'-'

```

例题:

<http://sustoj.com/JudgeOnline/problem.php?id=1948>

AC code:

```

1  #include <stdio.h> //c 版
2  #include <string.h>
3  #define maxn 100000
4  char ans[maxn];
5  char* strrev(char* s)
6  {
7      /* h指向s的头部 */
8      char* h = s;
9      char* t = s;
10     char ch;
11     /* t指向s的尾部 */
12     while(*t++){
13         t--; /* 与t++抵消 */
14         t--; /* 回跳过结束符'\0' */
15         /* 当h和t未重合时，交换它们所指向的字符 */
16         while(h < t)
17         {
18             ch = *h;
19             *h++ = *t; /* h向尾部移动 */
20             *t-- = ch; /* t向头部移动 */
21         }
22         return(s);
23     }
24     int judge(char *a, char *b){
25         int lena=strlen(a);
26         int lenb=strlen(b);
27         if(lena>lenb) return 1;
28         if(lena<lenb) return -1;
29         return strcmp(a,b);
30     }
31     const char *num_sub(char *a, char *b){
32         char aa[maxn],bb[maxn];
33         memset(aa,0,sizeof aa);
34         memset(bb,0,sizeof bb);
35         memset(ans,0,sizeof ans);
36         char *as=ans;
37         int cnt=judge(a,b);
38         bool bo=false;
39         if(cnt>0){
40             strcpy(aa,a);
41             strcpy(bb,b);
42             bo=false;
43         }else if(cnt<0){
44             strcpy(aa,b);
45             strcpy(bb,a);
46             bo=true;
47         }else{
48             ans[0]='0';
49             return as;
50         }
51         int lena=strlen(aa);
52         int lenb=strlen(bb);
53         strrev(aa);

```

```

54     strrev(bb);
55     int i;
56     for(i=0;;i++){
57         if(i<lena&& i<lenb){
58             if(aa[i]>=bb[i]){
59                 ans[i]=aa[i]-bb[i]+'0';
60             }else {
61                 ans[i]=aa[i]-bb[i]+10+'0';
62                 for(int j=i+1;;j++){
63                     if(aa[j]!='0') aa[j]='9';
64                     else{
65                         aa[j]--;
66                         break;
67                     }
68                 }
69             }
70         }else if(i<lena){
71             ans[i]=aa[i];
72         }else if(i<lenb){
73             ans[i]=bb[i];
74         }else break;
75     }
76     while(i--){
77         ans[i+1]='\0';
78         if(ans[i]-'0'>0){
79             break;
80         }
81     }
82     if(bo) ans[i+1]='-';
83     strrev(ans);
84     return as;
85 }
86 char a[maxn],b[maxn];
87 int main(){
88     while(~scanf("%s %s", a, b)){
89         printf("%s\n", num_sub(a,b));
90     }
91     return 0;
92 }
93

```

```

1  #include <bits/stdc++.h> //c++ 版
2  using namespace std;
3  #define maxn 100000
4  string ans;
5  int judge(string a,string b){
6      int lena=a.size();
7      int lenb=b.size();
8      if(lena>lenb) return 1;
9      if(lena<lenb) return -1;
10     if(a>b) return 1;
11     if(a==b) return 0;
12     if(a<b) return -1;
13 }
14 string num_mul(string a,string b){
15     string aa,bb;
16     aa.clear();
17     bb.clear();
18     ans.clear();
19     bool bo;
20     int cnt=judge(a,b);
21     if(cnt>0){
22         aa=a;
23         bb=b;
24         bo=false;
25     }else if(cnt<0){
26         aa=b;
27         bb=a;
28         bo=true;
29     }else if(cnt==0){

```

```

30     ans+='0';
31     return ans;
32 }
33 reverse(aa.begin(),aa.end());
34 reverse(bb.begin(),bb.end());
35 int lena=aa.size();
36 int lenb=bb.size();
37 int i;
38 for(i=0;;i++){
39     if(i<lena&&i<lenb){
40         if(aa[i]>=bb[i]){
41             ans+=aa[i]-bb[i]+'0';
42         }else{
43             ans+=aa[i]-bb[i]+10+'0';
44             for(int j=i+1;j++){
45                 if(aa[j]=='0') aa[j]='9';
46                 else {
47                     aa[j]--;
48                     break;
49                 }
50             }
51         }
52     }else if(i<lena){
53         ans+=aa[i];
54     }else if(i<lenb){
55         ans+=bb[i];
56     }else break;
57 }
58 while(i--){
59     if(ans[i]-'0'>0){
60         break;
61     }
62 }
63 ans.erase(i+1,lena-i);
64 if(!ans) ans+='-';
65 reverse(ans.begin(),ans.end());
66 return ans;
67 }
68 string a,b;
69 int main(){
70     while(cin>>a>>b){
71         cout<<num_mul(a,b)<<endl;
72     }
73     return 0;
74 }
75

```

大数乘法

第一步还是要最低位对齐

```

1  例如：
2  1 2 3 4 5 6 7 8 9
3  2 0 0 0 1 1 2 8
4  我是这样子在实现的
5  strrev(a)//char的反转函数
6  reverse(a.begin(),a.end());//string的反转函数
7  反转过来就是这个
8  9 8 7 6 5 4 3 2 1
9  8 2 1 1 0 0 0 2

```

第二步就是上下相乘，向后进位

```

1  9 8 7 6 5 4 3 2 1
2  8 2 1 1 0 0 0 2
3  -----
4  72 64 56 48 40 32 24 16 8-----8*(987654321)
5  18 16 14 12 10 8 6 4 2 -----2*(987654321)
6  9 8 7 6 5 4 3 2 1-----1*(987654321)
7  9 8 7 6 5 4 3 2 1-----1*(987654321)

```

```

8      0 0 0 0 0 0 0 0 0 0-----0*(987654321)
9      0 0 0 0 0 0 0 0 0 0-----0*(987654321)
10     0 0 0 0 0 0 0 0 0 0-----0*(987654321)
11     18 16 14 12 10 8 6 4 2 -----2*(987654321)
12 -----
13 72 82 81 79 67 55 43 49 35 21 15 11 8 6 4 2
14 -----
15 7 8 8 8 7 6 4 5 4 2 1 1
16 2 9 9 7 5 2 9 3 0 5 7 2 9 6 4 2
17 反转过来就是这样了
18 2469275039257992

```

例题

<http://sustoj.com/JudgeOnline/problem.php?id=1985>

AC code:

```

1  #include <stdio.h> //c 版
2  #include <string.h>
3  using namespace std;
4  #define maxn 500000
5  int ans[maxn];
6  char num[maxn];
7  char *strrev(char *str)
8  {
9      char *p1, *p2;
10     if (!str || !*str)
11     {
12         return str;
13     }
14     for (p1 = str, p2 = str + strlen(str) - 1; p2 > p1; ++p1, --p2)
15     {
16         *p1 ^= *p2;
17         *p2 ^= *p1;
18         *p1 ^= *p2;
19     }
20     return str;
21 }
22 char* num_mul(char *a, char *b){
23     memset(ans,0,sizeof ans);
24     memset(num,0,sizeof num);
25     char *as=num;
26     int nu=0;
27     memset(ans,0,sizeof ans);
28     int lena=strlen(a);
29     int lenb=strlen(b);
30     strrev(a);
31     strrev(b);
32     for(int i=0;i<lena;i++){
33         for(int j=0;j<lenb;j++){
34             ans[i+j]+=(a[i]-'0')*(b[j]-'0');
35         }
36     }
37     int i;
38     int yu=0;
39     int len=lena+lenb-1;
40     for(i=0;;i++){
41         if(i<len||yu!=0){
42             yu=ans[i]+yu;
43             ans[i]=yu%10;
44             yu=yu/10;
45             }else break;
46         }
47         while(i--){
48             num[nu++]=ans[i]+'0';
49         }
50         return num;
51     }
52     char a[maxn],b[maxn];

```

编程语言大PK，你选谁？

关闭

```

53 int main(){
54     while(~scanf("%s %s", a, b)){
55         if(strcmp(a,"0")==0) printf("0\n");
56         else if(strcmp(b,"0")==0) printf("0\n");
57         else printf("%s\n", num_mul(a,b));
58     }
59     return 0;
60 }
61

```

```

1  #include <cstring>
2  #include <cstdio>
3  #include <iostream>
4  #include <algorithm>
5  using namespace std;
6  #define maxn 100000
7  int ans[maxn];
8  string num;
9  string num_mul(string a,string b){
10     memset(ans,0,sizeof ans);
11     num.clear();
12     int lena=a.size();
13     int lenb=b.size();
14     reverse(a.begin(),a.end());
15     reverse(b.begin(),b.end());
16     for(int i=0;i<lena;i++){
17         for(int j=0;j<lenb;j++){
18             ans[i+j]+=(a[i]-'0')*(b[j]-'0');
19         }
20     }
21     int yu=0;
22     int i;
23     int len=lena+lenb-1;
24     for(i=0;;i++){
25         if(i<len||yu!=0){
26             yu=ans[i]+yu;
27             ans[i]=yu%10;
28             yu=yu/10;
29             }else break;
30     }
31     while(i--){
32         num+=ans[i]+'0';
33     }
34     return num;
35 }
36 string a,b;
37 int main(){
38     while(cin>>a>>b){
39         cout<<num_mul(a,b)<<endl;
40     }
41     return 0;
42 }
43
44

```

大数除法

已经学会大数减法了，那就不用再去模拟大数除法了

可以一直减下去用大数减了多少步，但这样会有缺陷，比如说

$\frac{10000000000}{1}$ 那就要跑10000000000次，这样肯定会超时的

那可以换一种方法

比如说 $\frac{a}{b}$

先给分母 b 末尾补零直到与 a 长度相同，得到 b' ，

比较 a 和 b' 的大小，大的话一直减，减到 a 小于 b' ，然后再给 b' 去掉末尾的零

一直循环上面布置，直到 b' 还原到 b 的时候停止

这样可能会出现前置零，那就去掉前置零就好了

比如

$\frac{10000000000}{1}$ 那就跑11次就好了

例题

<http://sustoj.com/JudgeOnline/problem.php?id=1986>

AC code :

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define maxn 1000
4  char ans[maxn];
5  char ans1[maxn];
6  char *strrev(char *str)
7  {
8      char *p1, *p2;
9      if (! str || ! *str)
10     {
11         return str;
12     }
13     for (p1 = str, p2 = str + strlen(str) - 1; p2 > p1; ++p1, --p2)
14     {
15         *p1 ^= *p2;
16         *p2 ^= *p1;
17         *p1 ^= *p2;
18     }
19     return str;
20 }
21 int judge(char *a, char *b){
22     int lena=strlen(a);
23     int lenb=strlen(b);
24     if(lena>lenb) return 1;
25     if(lena<lenb) return -1;
26     return strcmp(a,b);
27 }
28 const char *num_sub(char *a, char *b){
29     char aa[maxn],bb[maxn];
30     memset(aa,0,sizeof aa);
31     memset(bb,0,sizeof bb);
32     memset(ans,0,sizeof ans);
33     char *as=ans;
34     int cnt=judge(a,b);
35     bool bo=false;
36     if(cnt>0){
37         strcpy(aa,a);
38         strcpy(bb,b);
39         bo=false;
40     }else if(cnt<0){
41         strcpy(aa,b);
42         strcpy(bb,a);
43         bo=true;
44     }else{
45         ans[0]='0';
46         return as;
47     }
48     int lena=strlen(aa);
49     int lenb=strlen(bb);
50     strrev(aa);
51     strrev(bb);
52     int i;
53     for(i=0;;i++){
54         if(i<lena&&i<lenb){
55             if(aa[i]>bb[i]){
56                 ans[i]=aa[i]-bb[i]+'0';
57             }else {
58                 ans[i]=aa[i]-bb[i]+10+'0';
59                 for(int j=i+1;j++){
60                     if(aa[j]=='0') aa[j]='9';
61                     else{
62                         aa[j]--;
63                         break;
64                     }
65                 }
66             }
67         }
68     }
```

```

66     }
67     }else if(i<lena){
68         ans[i]=aa[i];
69     }else if(i<lenb){
70         ans[i]=bb[i];
71     }else break;
72     }
73     while(i--){
74         ans[i+1]='\0';
75         if(ans[i]!='0'){
76             break;
77         }
78     }
79     if(b0) ans[i+1]='-';
80     strrev(ans);
81     return as;
82 }
83 char *num_div(char *a, char *b){
84     memset(anss,0,sizeof anss);
85     int nu=0;
86     int lena=strlen(a);
87     int lenb=strlen(b);
88     for(int i=lenb;i<lena;i++){
89         b[i]='\0';
90     }
91     while(lena-->lenb){
92         int i=0;
93         while(judge(a,b)>=0){
94             i++;
95             strcpy(a,num_sub(a,b));
96         }
97         anss[nu++]=i+'0';
98         b[lena]='\0';
99     }
100     if(anss[0]=='\0'){
101         anss[0]='0';
102         return anss;
103     }
104     if(anss[0]=='0')
105         return anss+1;
106     return anss;
107 }
108 char a[maxn],b[maxn];
109 int main(){
110     while(~scanf("%s %s", a, b)){
111         printf("%s\n", num_div(a,b));
112         cout<<a<<endl;
113     }
114 }

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define maxn 100000
4  string ans;
5  string anss;
6  string yu;
7  int judge(string a,string b){
8      int lena=a.size();
9      int lenb=b.size();
10     if(lena>lenb) return 1;
11     if(lena<lenb) return -1;
12     if(a>b) return 1;
13     if(a==b) return 0;
14     if(a<b) return -1;
15 }
16 string num_mul(string a,string b){
17     string aa,bb;
18     aa.clear();
19     bb.clear();
20     ans.clear();

```

```
21 bool bo;
22 int cnt=judge(a,b);
23 if(cnt>0){
24     aa=a;
25     bb=b;
26     bo=false;
27 }else if(cnt<0){
28     aa=b;
29     bb=a;
30     bo=true;
31 }else if(cnt==0){
32     ans+='0';
33     return ans;
34 }
35 reverse(aa.begin(),aa.end());
36 reverse(bb.begin(),bb.end());
37 int lena=aa.size();
38 int lenb=bb.size();
39 int i;
40 for(i=0;;i++){
41     if(i<lena&&i<lenb){
42         if(aa[i]>=bb[i]){
43             ans+=aa[i]-bb[i]+'0';
44         }else{
45             ans+=aa[i]-bb[i]+10+'0';
46             for(int j=i+1;j++){
47                 if(aa[j]=='0') aa[j]='9';
48                 else {
49                     aa[j]--;
50                     break;
51                 }
52             }
53         }
54     }else if(i<lena){
55         ans+=aa[i];
56     }else if(i<lenb){
57         ans+=bb[i];
58     }else break;
59 }
60 while(i--){
61     if(ans[i]-'0'>0){
62         break;
63     }
64 }
65 ans.erase(i+1,lena-i);
66 if(bo) ans+="-";
67 reverse(ans.begin(),ans.end());
68 return ans;
69 }
70 string num_div(string a,string b){
71     anss.clear();
72     int lena=a.size();
73     int lenb=b.size();
74     for(int i=lenb;i<lena;i++){
75         b+='0';
76     }
77     while(lena-->=lenb){
78         int i=0;
79         while(judge(a,b)>=0){
80             a=num_mul(a,b);
81             i++;
82         }
83         anss+=(i+'0');
84         b.erase(lena);
85     }
86     yu=a;
87     if(anss.size()==0){
88         anss+='0';
89         return anss;
90     }
91     if(anss[0]=='0')
```

```
92         anss.erase(0,1);
93     return anss;
94 }
95 string a,b;
96 int main(){
97     while(cin>>a>>b){
98         cout<<num_div(a,b)<<endl;
99         cout<<yu<<endl;
100     }
101     return 0;
102 }
103
```

有 0 个人打赏

文章最后发布于: 2019

编程语言大PK，你选谁？

关闭