Tháng Tư 14, 2015

Bài toán tính tổng các số nguyên rất lớn

4 phản hồi

1. Giới thiệu:

– Trong một vài trường hợp, bạn cần lưu một số nguyên cực kỳ lớn.

– Chẳng hạn: 123.456.789.123.456.789.123.456.789.123.456.789.123.456.789.123.456.789

– Bạn có bao giờ thắc mắc có kiểu dữ liệu nào có thể đáp ứng được chuyện đó?

– Kiểu int chỉ có thể biểu diễn một số nguyên khoảng 2,1 tỉ. Ngoài ra, chúng ta còn có kiểu "unsigned long long int" với việc có biểu diễn một số có 18 ký số. Trong đó, từ khóa "unsigned" nhằm mục đích chỉ biểu diễn số dương. Nếu không có từ khóa "unsigned" thì vấn đề gì xảy ra, bạn tự tìm hiểu nhé!

– Trong bài này, tôi sẽ hướng dẫn các bạn thực hiện các phép tính cơ bản + – * / giữa 2 " **số nguyên**". Nhưng "**số nguyên**" mà tôi muốn nói là số rất lớn, có số lượng ký số lên đến hàng chục, hàng ngàn và thậm chí là hàng tỉ ký số.

2. Hiện thực:

– Để sử dụng lớp vector, bạn cần thư viện <vector> và khai báo using namespace std

Hình 1 – Khởi tạo và xem kết quả sau khi khởi tạo

```cpp
int main()
{
    char a[] = "00125";
    char b[] = "0045";

    vector<int> v1, v2, v3;

    //khởi tạo 2 số
    init(a, v1);
    init(b, v2);

    //in 2 số ra màn hình
    print(v1);
    cout << endl;
    print(v2);
    cout << endl;
    return 0;
}
```

Hình 1

Hình 2 – Hàm khởi tạo một số

```cpp
//vector khác với mảng
//tham số vector phải là một tham chiếu
//nếu không nó chỉ là một vector bảng sao
void init(char str[], vector<int> &v)
{
    for (int i = strlen(str) - 1; i >= 0; i--)
    {
        //nên chuyển toàn bộ sang kiểu int
        //vì thường xuyên tính toán
        //nếu gặp kiểu char cùng với 1 phép tính
        //thì chương trình sẽ tự động chuyển kiểu
        //vì vậy ta không nên sử dụng mảng char
        v.push_back(str[i] - '0');
    }
    removeZero(v);
}
```

Hình 2

Hình 3 – Hàm loại bỏ các số 0 ở đầu

```cpp
//hàm loại bỏ toàn bộ số 0 đầu tiên
//0123 -> 123
//đây là LÝ DO vì sao chúng ta khởi tọa một vector ngược
void removeZero(vector<int> &v)
{
    while (v[v.size() - 1] == 0     //trong khi số đầu là số 0
           && v.size() > 1)         //và số ký số vẫn lớn hơn 1
        v.pop_back();               //loại bỏ số 0 đó
}
```

Hình 3

Hình 4 – Hàm in số nguyên

```cpp
void print(vector<int> v)
{
    for (int i = v.size() - 1; i >= 0; i--)
        cout << v[i];
}
```

Hình 4

– Sau đó, các bạn thử khởi tạo 2 số và in ra màn hình nhé.

*– Bây giờ chúng ta sẽ bắt đầu vào thực hiện các phép tính + – * /*

Hình 5 – Phép cộng

```
void add(vector<int> v1, vector<int> v2, vector<int> &v3)
{
    //khởi tạo v3 là rỗng
    v3.clear();

    //tìm ra số lớn hơn -> gán các số 0 ở đầu cho số nhỏ hơn
    int length = v1.size() > v2.size() ? v1.size() : v2.size();
    //gán các số 0 ở đầu
    v1.resize(length);
    v2.resize(length);

    //số dư
    int extra = 0;
    int temp;
    for (int i = 0; i < length; i++)
    {
        temp = v1[i] + v2[i] + extra;
        v3.push_back(temp % 10);
        extra = temp / 10;
    }
    //số dư cuối cùng
    if (extra > 0)
        v3.push_back(temp);
}
```

Hình 5

Hình 6 – Phép trừ

```
//phép trừ: chỉ xử lý trường hợp v1 >= v2
void sub(vector<int> v1, vector<int> v2, vector<int> &v3)
{
    //khởi tạo v3 là rỗng
    v3.clear();

    int length = v1.size();
    //gán các số 0 ở đầu
    v2.resize(length);

    //số dư
    int extra = 0;
    int temp;
    for (int i = 0; i < length; i++)
    {
        //trường hợp số bị trừ lớn hơn
        //tiến hành nhớ 1
        if (v1[i] < v2[i] + extra)
        {
            v3.push_back(v1[i] - v2[i] - extra + 10);
            extra = 1;
        }
        //trường hợp số bị trừ nhỏ hơn
        else
        {
            v3.push_back(v1[i] - v2[i] - extra);
            extra = 0;
        }
    }
    //bỏ các số 0 ở đầu
    removeZero(v3);
}
```

Hình 6

Hình 7 – Phép nhân

```
void mul(vector<int> v1, vector<int> v2, vector<int> &v3)
{
    //khởi tạo v3 là rỗng
    v3.clear();

    //chúng ta cần đến phép +
    //cần định nghĩa phép + trước khi làm phép nhân

    //duyệt từng hàng đơn vị của v1
    for (int i = 0; i < v1.size(); i++)
    {
        //tạo một vector tạm
        vector<int> v4;

        //số dư
        int extra = 0, temp;

        //duyệt từng hàng đơn vị của v2
        //->v1[i] * v2
        for (int j = 0; j < v2.size(); j++)
        {
            //phép * sẽ tương tự phép +
            temp = v1[i] * v2[j] + extra;
            v4.push_back(temp % 10);
            extra = temp / 10;
        }
        if (extra > 0)
            v4.push_back(extra);
        //gán i số 0 ở cuối
        for (int j = 0; j < i; j++)
             v4.insert(v4.begin(), 0);

        add(v3, v4, v3);
    }
}
```

Hình 7

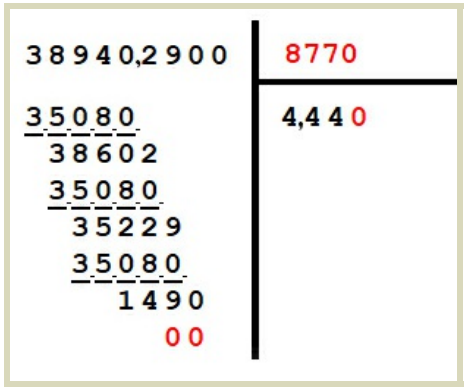
Hình 10 – Phép chia

– Trong 4 phép toán thì phép chia là phép dài dòng và phức tạp nhất

– Phép chia cần đến hàm so sánh và phép trừ

Hình 8 – Thực hiện phép chia thủ công

– Chúng ta sẽ thực hiện việc chia 2 số như cách mà chúng ta đã biết từ khi học cấp 1



Hình 8

Hình 9 – Phép so sánh

```
//thực hiện so sánh giữa 2 số nguyên
//a > b => -1
//a < b => 1
//a == b => 0
int comepare(vector<int> a, vector<int> b)
{
    //so sánh theo chiều dài
    if (a.size() > b.size())
        return -1;
    if (a.size() < b.size())
        return 1;
    for (int i = a.size() - 1; i >= 0; i--)
    {
        if (a[i] > b[i])
            return -1;
        if (a[i] < b[i])
            return 1;
    }
    return 0;
}
```

Hình 9

```
void div(vector<int> a, vector<int> b, vector<int> &c)
{
    //khởi tạo c là rỗng
    c.clear();

    //số sẽ bị tách khỏi a
    vector<int> carry;

    //duyệt hết a
    while (a.size() > 0)
    {
        //tính vị trí bắt đầu và kết thúc của số a cần tách
        int finish = a.size() - 1;
        int start = finish - b.size() + 1;
        if (start < 0)
            start = 0;
        int count = start;

        //gán vào carry
        if (carry.size() == 0)
            while (count <= finish)
            {
                carry.push_back(a[start]);
                a.pop_back();
                count++;
            }
        else
            while (count <= finish)
            {
                carry.insert(carry.begin(), a[start]);
                a.pop_back();
                count++;
            }

        //nếu carry chưa đủ lớn thì tách thêm 1 ký số của a
        if (comepare(carry, b) != -1 && a.size() > 0)
        {
            carry.insert(carry.begin(), a[start - 1]);
            a.pop_back();
        }

        //xóa các số 0 ở đầu sau khi tách
        removeZero(carry);

        //thực hiện việc chia carry cho b
        count = 0;
        while (comepare(carry, b) != 1)
        {
            count++;
            sub(carry, b, carry);
        }

        //c là kết quả cần tìm
        if (count >= 10)
        {
            int x = count % 10;
            count /= 10;
            int y = count % 10;
            count /= 10;
            c.insert(c.begin(), y);
            c.insert(c.begin(), x);
        }
        else
            c.insert(c.begin(), count);
    }
}
```

Hình 10

Như vậy là tôi vừa hướng dẫn các bạn thực hiện các phép tính cơ bản giữa các số nguyên cực lớn. Các bạn hãy thử cải tiến các thuật toán trên bằng cách mỗi phần tử của 1 vector có thể lưu 9 ký số chứ không phải là 1 ký số như tôi đã trình bày. Nếu được, hãy kiểm tra tính đúng đắn của thuật toán và liên hệ tôi để cùng nhau thảo luận vấn đề này nhé

Liên quan

Lập trình Windows - Truyền tham số giữa 2 Dialog
In "Windows programing C++"

HTML5 - Làm game trên Windows 8
In "HTML5"

Windows Form - Tạo một giao diện có thể vẽ
In "Windows progamming C#"

Posted by Chuong Le Hoang in Data structures & Algorithms

← Thuật toán Eratosthenes – Tối ưu thuật toán sàng số nguyên tố

Thuật toán Kadane – Tìm tập hợp con có tổng lớn nhất →

4 thoughts on "Big Number – Các thao tác cộng trừ nhân chia số nguyên lớn"

Anonymous
Tháng Mười Một 2, 2016 lúc 10:12 chiều

vậy cho e hỏi làm sao để in ra được màn hình ạ

Phản hồi

Chuong Le Hoang
Tháng Mười Một 3, 2016 lúc 10:03 sáng

Hi bạn, có function print(vector v) mình có hướng dẫn cài đặt trong bài viết, bạn tham khảo nhé.
Chúc bạn thành công!

Phản hồi

Nguyễn Minh
Tháng Mười Một 15, 2016 lúc 10:06 chiều

bạn ơi bạn có thể dùng cách cách này mà cài đặt trên mảng dc k ạ,mình thực sự đang rất cần ạ

Nguyễn Minh
Tháng Mười Một 15, 2016 lúc 10:03 chiều

bạn có thể hướng dẫn sơ qua về phần này mà bằngmảng được không ạ,mình đang rất rất rất cần ạ,nếu được bạn có thể reply mình,mình cảm ơn

Phản hồi

Trả lời

Nhập bình luận của bạn tại đây...