Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

Тульский государственный университет

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

РЕКУРРЕНТНОЕ УМНОЖЕНИЕ ДЛИННЫХ ЧИСЕЛ

Лабораторная работа № 5
по курсу «Структуры и алгоритмы обработки данных»

Вариант № 4

Выполнил: студент группы 220601 _________ Белым А.А.
(подпись)
Проверил: д. ф.-м.н, проф.каф. ИБ _________ Двоенко С.Д.
(подпись)

Тула 2013

Цель работы

Изучить метод Карацубы для рекуррентного умножения больших чисел.

Задание

Напишите функцию, выполняющую действие: A*C*C

В качестве тестового примера примените :

A = 111111111111112222222222222223333333333333

C = 444444444444446666666666666661111111111111

Теоретическая справка

Перемножение двух n-значных целых чисел обычным школьным методом «в столбик» сводится, по сути, к сложению n n-значных чисел. Поэтому для сложности этого «школьного» или «наивного» метода имеем оценку сверху:

$$M(n) = O(n^2).$$

В 1956 году А. Н. Колмогоров сформулировал гипотезу, что нижняя оценка для $M(n)$ при любом методе умножения есть также величина порядка n^2 (так называемая «гипотеза n^2» Колмогорова). На правдоподобность гипотезы n^2 указывал тот факт, что метод умножения «в столбик» известен не менее четырёх тысячелетий (например, этим методом пользовались шумеры), и если бы был более быстрый метод умножения, то он, вероятно, уже был бы найден. Однако, в 1960 году Анатолий Карацуба нашёл новый метод умножения двух n-значных чисел с оценкой сложности

$$M(n) = O(n^{\log_2 3})$$

и тем самым опроверг «гипотезу n^2».

Впоследствии метод Карацубы был обобщён до парадигмы «разделяй и властвуй», другими важными примерами которой являются метод двоичного разбиения ,двоичный поиск, метод бисекции и др.

Рассмотрим метод Карацубы. Пусть x и y — два $2n$-значных десятичных числа:

$$x = (x_{2n-1},\ldots, x_0),\ y = (y_{2n-1},\ldots, y_0),\ x_t, y_t = 0,1,\ldots 9.$$

Представим их в виде $x = 10^n a + b, y = 10^n c + d$, где a, b, c, d- n-значные числа. Применив тождество $(b - a)\cdot(c - d) = -bd - ac + bc + ad$, получим:

$$xy = (10^n a + b)(10^n c + d) = (10^{2n} + 10^n)bd + 10^n(b - a)(c - d) + (10^n + 1)ac\,.$$

Задача умножения $2n$-разрядных чисел оказалась сведена к трем операциям для n-разрядных чисел и к операциям сдвига.

Может показаться, что выигрыш по сравнению с обычным умножением незначителен — только в ¾ раза. Но и сами n-значные числа можно умножать таким же образом. Поэтому множитель ¾ будет возникать при каждом удвоении числа разрядов. Тогда, например, при умножении 1024-значных чисел накопится более чем десятикратный выигрыш.

Умножение Карацубы на n-значных числах будет эффективнее умножения в столбик в $(3/4)^{\log_2 n}$ раза. Это дает выигрыш в $n^{\log_2 3} \approx n^{1,6}$ раз.

Текст программы

Далее представлен текст программы на языке C++, реализующей вычисление произведения длинных чисел.

```
#include <iostream>
#include <cstdio>
#include <sstream>
#include <iomanip>
typedef unsigned long long uint;

const uint SIZE=(sizeof(uint))*8,
           SIZE2=SIZE/2,
           POW2=uint(1)<<SIZE2;
const uint MAX_SIZE=10000;
using namespace std;
inline void addc(uint& a,const uint& b,uint& carry){
    a+=b;
    if(a<b)
        ++carry;
}

void long_len_correct(uint *src,uint &len){
    for(uint *ptr=src+len-1;!(*ptr);ptr--,len--);
}
void long_clear(uint *src,uint len){
    for(;len;src++,len--)
        *src=0;
}

void mult1(uint a,uint b,uint *res,uint& car){
    uint r0=(a%POW2)*(b%POW2),
         r01=(a%POW2)*(b>>SIZE2),
```

```
            r10=(a>>SIZE2)*(b%POW2),
            r1=(a>>SIZE2)*(b>>SIZE2);
    uint car0=0;
    addc(res[0],r0,car0);
    addc(res[0],(r01%POW2)<<SIZE2,car0);
    addc(res[0],(r10%POW2)<<SIZE2,car0);

    addc(res[1],car0,car);
    addc(res[1],r1,car);
    addc(res[1],r01>>SIZE2,car);
    addc(res[1],r10>>SIZE2,car);
}

void long_mult(uint *a, uint la,uint *b, uint lb, uint *res,uint &len){
    uint i,j,car0=0,car1=0,car2=0,maxi=la+lb-1;
    long_clear(res,la+lb);
    for(i=0;i<=maxi;i++){
        addc(res[i],car0,car1);
        for(j=0;j<=i;j++){
            if(j>=la)
                break;
            if((i-j)>=lb)
                continue;
            mult1(a[j],b[i-j],res+i,car2);
        }
        car0=car1;car1=car2;car2=0;
    }
    len=maxi+1;
    if(car0){
        len=maxi+1;
        res[len]=car0;
        len++;
    }
    if(car1){
        len=maxi+2;
        res[len]=car1;
        len++;
    }
    if(car2){
        len=maxi+3;
        res[len]=car2;
        len++;
    }
    long_len_correct(res,len);
}

void print_long(uint* num,uint len){
    for(uint *ptr=num+len-1;len;--ptr,--len){
        cout<<hex<<setw(sizeof(uint)*2)<<setfill('0')<<*ptr;
    }
    cout<<endl;
}

void long_from_string(const string &str,uint *res,uint &len){
    uint block_size=SIZE/4;
    int i; len=0;
    for(i=str.length()-block_size;i>=0;i-=block_size){
        istringstream istr(str.substr(i,block_size));
        istr>>hex>>*res;
        len++;res++;
    }
    if((i+block_size)>0){
        istringstream istr(str.substr(0,i+block_size));
        istr>>hex>>*res;
        len++;res++;
    }
```

```
}

void input_long(uint *res,uint &len){
    string s;
    cin>>s;
    long_clear(res,len);
    long_from_string(s,res,len);
}

int main()
{
    uint a[MAX_SIZE],c[MAX_SIZE],a_len=MAX_SIZE,c_len=MAX_SIZE;
    cout<<"Программа вычисляет A*C*C."<<endl;
    cout<<"Введите A."<<endl;
    input_long(a,a_len);
    cout<<"Введите C."<<endl;
    input_long(c,c_len);
    uint ac[MAX_SIZE],acc[MAX_SIZE],ac_len,acc_len;
    long_mult(a,a_len,c,c_len,ac,ac_len);
    long_mult(ac,ac_len,c,c_len,acc,acc_len);
    cout<<"A*C*C ="<<endl;
    print_long(acc,acc_len);
    return 0;
}
```

Тестовый пример

На рисунке 1 представлен пример работы программы, реализующей произведение длинных чисел.

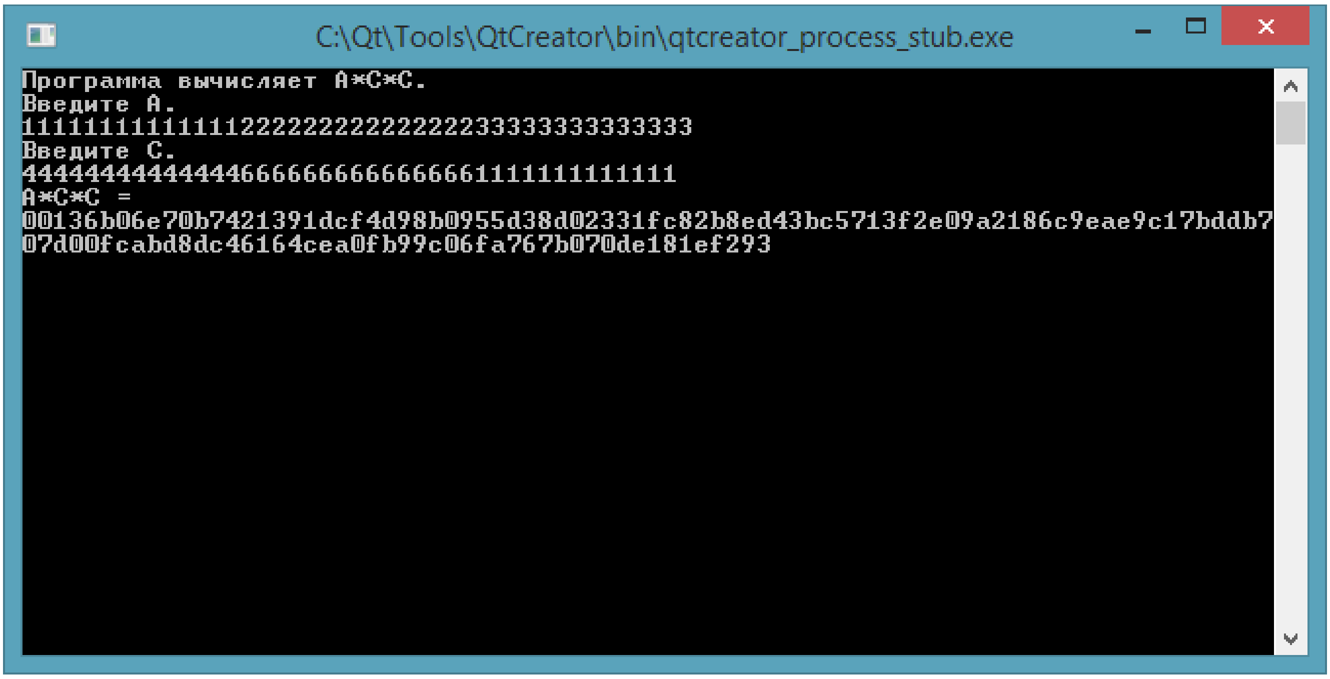

Рисунок 1 — Пример работы программы вычисления произведения длинных чисел

Вывод

В данной работе я познакомился с перемножением длинных чисел методом Карацубы. Данный метод позволяет существенно ускорить умножение по сравнению с обычным методом в столбик. Была написана программа, реализующая умножения методом Карацубы.