**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе № 1**

**по дисциплине «Программирование»**

**Тема: Рекурсия**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Азаревич А.Д. |
| Преподаватель |  | Шолохова О.М. |

Санкт-Петербург

2017

**Цель работы.**

Ознакомиться с основными понятиями и приёмами рекурсивного программирования, получить навыки программирования рекурсивных процедур и функций на языке программирования C++.

**Постановка задачи.**

*Требования и рекомендации* к выполнению задания:

1. проанализировать полученное задание, выделив рекурсивно определяемые информационные объекты и (или) действия;
2. разработать программу, использующую рекурсию;
3. сопоставить рекурсивное решение с итеративным решением задачи;
4. сделать вывод о целесообразности и эффективности рекурсивного решения данной задачи.

Задание:

Для заданных неотрицательных целых *n* и *m* вычислить (рекурсивно) биномиальные коэффициенты, пользуясь их определением:



**Основные теоретические положения.**

Согласно поставленной задаче



а так же m и n являются целыми числами, при этом оба больше 0. Т.к. биномиальный коэффициент есть число сочетаний из n по m, то из формулы

получим, что условие при m>n0 является скорее ограничением на вводимые данные. Поэтому это условие в рекурсивную функцию можно не вставлять, делая данную проверку перед первым вызовом функции.

Попробуем упростить условие при m=0, n>0 или m=n0. Согласно выражению часть условия про n>0 неактуальна, т.к. сначала «обнулится» m (m<n, на «m-мном» шаге m=m-m=0, n=n–m>0). В случае же m=n .

Перепишем условие с этими правками:



n m 0

Рекурсивная функция будет выглядеть так:

F(m, n) = F(m-1, n-1)+F(m, n-1)

F(0, k) = F(k, k) = 1

k принадлежит промежутку [0, n].

Однако, данная рекурсия будет очень невыгодна по времени, т.к. некоторые элементы могут пересчитываться по многу раз. Для того, чтобы ускорить программу, мы будем записывать уже вычисленные коэффициенты в специальную таблицу готовых результатов.

**Спецификация программы.**

*Назначение программы*.

Программа предназначена для вычисления биномиальных коэффициентов с помощью рекурсии.

*Описание программы*.

Программа написана на языке С++ с использованием компилятора ml64 9.0.30729.1. Входными данными для программы являются 2 целых числа, больших 0, вводимые пользователем с клавиатуры или загружаемые из файла test.txt. Проверка значений, вводимых пользователем, отсутствует, поведение программы при некорректном вводе не определено. Выходными данными является число, равное числу сочетаний (биномиальному коэффициенту) от заданных чисел.

*Пример диалога с пользователем*.

Найден файл с тестами

m= 5; n= 6; Бин. коэф.= 6

m= 3; n= 4; Бин. коэф.= 10

m= 10; n= 12; Бин. коэф.= 76

m= 12; n= 12; Бин. коэф.= 77

m= 0; n= 5; Бин. коэф.= 78

m= 5; n= 0; Бин. коэф.= 78

m= 1; n= 13; Бин. коэф.= 91

Прогон тестового файла завершён

m= 5

n= 10

Бин. коэф.= 343

m= 15

n= 20

Бин. коэф.= 15847

*Реализация*.

Классы:

1. Binomial – класс, хранящий в себе все вычисленные биномиальные коэффициенты, а так же использующийся для их рекурсивного вычисления.

Переменные класса:

1. unsigned long long \*\*save - массив указателей на массивы со всеми возможными значениями биномиальными коэффициентами для заданого n.
2. unsigned int Nmax – условное количество указателей, хранящихся в save. Равно максимально встретившемуся n.
3. bool NotOverFlow – флаг переполнения: равен true, если вычесленный биномиальный коэффициент от заданных m и n меньше макимального допустимого значения для unsigned long long, и false в противном случае.

Функции класса:

1. b\_k();

Конструктор класса. Выделяет память под save, заполняет таблицу коэффициентов для n=2 и устанавливает Nmax = 2.

1. ~b\_k();

Деструктор класса. Освобождает всю выделенную под хранение вычисленных коэффициентов память.

1. void more(ui n);

Расширяет хранилище вычисленных коэффициентов от Nmax до n.

Входные параметры:

Значение n, которое больше Nmax.

Выходные параметры отсутствуют.

1. ull binom(ui m, ui n);

Вычисляет значение биномиального коэффициента для заданных m и n.

Входные параметры:

значения m и n (коэффициенты в формуле)

Выходные параметры:

флаг переполнения (если возвращается false – произошло переполнение бин. коэф-та, если true – всё хорошо)

Возвращаемое значение:

Значение биномиального коэффициента для заданных m и n.

Предусловие:

m >= 0, n >= 0

Постусловие:

* Если m=0 или m=n возвращается 1;
* Если в нужной ячейке хранится 0, то начинаем вычислять по приведённой выше формуле. Если при вычислении данного коэффициента флаг isOverFlow примет значение folse в данную ячейку помещается 1;
* Если коэффициент в нужной ячейке равен 1, isOverFlow принимает значение folse;
* Если в нужной ячейке хранится знаение отличное от 0 или 1 возвращается данное значение.

1. bool colculate(ui &m, ui &n, ull &res)

«Отдаёт» пользователю значение биномиального коэффициента от m и n.

Входные параметры:

значения m и n (коэффициенты в формуле), место, куда будет записываться результат вычислений res;

Выходные параметры:

Вычисленный коэффициент res;

Возвращаемое значение:

Флаг NotOverFlow – false – произошло переполнение бин. коэффициента, если true – всё хорошо;

Побочные действия:

В начале работы функция устанавливает isOverFlow в состояние true.

**Тестирование.**

|  |  |  |
| --- | --- | --- |
| M | N | Результат |
| 0 | 0 | 1 |
| 0 | 5 | 1 |
| 5 | 0 | 0 |
| 5 | 6 | 6 |
| 5 | 10 | 252 |
| 10 | 10 | 1 |
| 15 | 10 | 0 |
| 10 | 37 | 348330136 |
| 10 | 50 | 10272278170 |

**Выводы.**

В ходе выполнения лабораторной работы были получены навыки в реализации функций, основанных на рекурсии.

Так же было выявлено, что рекурсивные функции могут оказаться весьма полезными, однако в некоторых случаях (когда одно и то же вычисляется по многу раз, как в данной задаче) время выполнения программы увеличивается неоправданно быстро. Для ускорения вычислений используется так называемое «динамическое программирование» смысл которого заключается в том, что мы сохраняем уже вычисленные значения. Это хорошо ускоряет нашу программу, однако много требовательней к объёму оперативной памяти. Из этого следует, что некоторые задачи следует решать итеративно, либо перед решением внимательно исследовать поставленную задачу в целях выявления более приемлемого её решения.

**Приложение A. Исходный код.**

b\_k.h

#include <iostream>

#include <fstream>

//#include <stdin.h>

//данного файла нет в VS2008

typedef unsigned long long ull;

typedef unsigned int ui;

class Binomial

{

public:

Binomial();

~Binomial();

bool colculate(ui &m, ui &n, ull &res);

private:

ull \*\*save; //хранение выч-ных бин. коэф.

bool NotOverFlow;//флаг переполнения

ui Nmax; //макc. встретившееся n

void more(ui n);

ull binom(ui m, ui n);

};

b\_k.cpp

#include "b\_k.h"

Binomial::Binomial()

{

save = (ull \*\*)malloc(sizeof(ull \*) );

Nmax = 2;

save[0] = (ull \*)malloc(sizeof(ull) );

save[0][0] = 2;

}

Binomial::~Binomial()

{

for (ui i=0; i < Nmax-1; i++)

free(save[i]);

free(save);

}

void Binomial::more(ui n)

{

save = (ull \*\*)realloc(save, (n-1)\*sizeof(ull \*) );

if (!save)

{

std::cerr << "Ошибка: память не была выделена\n";

system ("pause");

exit(1);

}

for (ui i=Nmax-1; i<(n-1); i++)

{

save[i] = (ull \*)calloc(i+1, sizeof(ull) );

if (!save[i])

{

std::cerr << "Ошибка: память не была выделена\n";

system ("pause");

exit(1);

}

}

Nmax = n;

}

ull Binomial::binom(ui m, ui n)

{

if ( (!m) || (m==n) ) return 1;

if ( !save[n-2][m-1] )

{

save[n-2][m-1]= binom(m-1, n-1);

if(!NotOverFlow)

return 1;

ull c = binom(m, n-1);

if ( (ULLONG\_MAX - c < save[n-2][m-1]) || (!NotOverFlow) )

{

NotOverFlow = false;

save[n-2][m-1] = 1;

return 1;

}

save[n-2][m-1] += c;

return save[n-2][m-1];

}

if ( save[n-2][m-1]==1 )

NotOverFlow = false;

return save[n-2][m-1];

}

bool Binomial::colculate(ui &m, ui &n, ull &res)

{

NotOverFlow = true;

if ( m > n )

{

res = 0;

return NotOverFlow;

}

if (n >= Nmax) more(n);

res = binom(m, n);

return NotOverFlow;

}

main.cpp

//

// Лабораторная работа №1

// main.cpp

// Азаревич артём, группа 6383

// 18.09.2017

//

// Программа для вычисления биномиальных коэффициентов

#include "b\_k.h"

using namespace std;

void f(ui &m, ui &n, Binomial &koef);

void main()

{

setlocale(0, "");

Binomial koef;

ui m, n;

//Прогон по тестовым данным из файла test.txt

ifstream fin ("test.txt");

if ( fin.is\_open() )

{

cout << "Найден файл с тестами\n\n";

while ( !fin.eof() )

{

fin >> m;

fin >> n;

cout << "m= " <<m<<"; n= "<<n<<"; ";

f(m, n, koef);

}

cout <<"Прогон тестового файла завершён\n\n\n";

fin.close();

}

while (true)

{

cout << "m= ";

cin >> m;

cout << "n= ";

cin >> n;

f(m, n, koef);

}

}

void f(ui &m, ui &n, Binomial &koef)

{

ull res = 0;

if ( koef.colculate(m, n, res) )

cout << "Бин. коэф.= "<<res << endl;

else

cout << "Произошло переполнение\n";

}