Министерство Образования и Науки Российской Федерации  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

**Курсовая работа**

**по практикуму на ЭВМ: структуры данных и алгоритмы**

Факультет: прикладной математики и информатики

Группа: ПМ-53

Студент: Тябин Егор Алексеевич

Преподаватель: Тракимус Юрий Викторович

### Новосибирск 2016

# Условие задачи

Подсчитать количество попарно неизоморфных графов с n вершинами и четырьмя ребрами.

# 2. Анализ задачи

2.1. Исходные данные задачи: *n* – количество вершин, *n∈N.*  
2.2. Результат: *result ∈ N –* количество попарно не изоморфных графов.

2.3. Решение.   
Для начала определим, какие графы будут использоваться в нашей задаче. Воспользуемся определением графа. Граф - это пара *(V, E),* где V-конечное непустое множество вершин, а E – множество неупорядоченных пар ˂u, w˃ вершин из V, называемых ребрами. То есть граф — это множество точек и множество линий, соединяющих эти точки. Ориентированный граф, или орграф, G= (V, E) отличается от графа тем, что E - это множество упорядоченных пар (u, w) вершин, u, w∈V, называемых дугами. Дуга (u, w) ведет от вершины u к вершине w. При этом вершину w называют преемником вершины u, а u – предшественником вершины w.  
Как видно, граф и ориентированный граф отличаются, что говорит о том, что в графе нет ориентированных линий. Значит в нашей задаче будут использоваться только графы без ориентированных линий. Но графы, удовлетворяющие нашей задачи, могут содержать разные цепи, петли(псевдограф), циклы.   
Для решения задачи нам понадобится определение изоморфного графа.  
Два графа *G*1 и *G*2 *изоморфны*, если существует взаимно однозначное отображение множества вершин графа *G*1 на множество вершин графа *G*2, сохраняющее смежность.

По условию задачи надо посчитать максимально возможное количество не изоморфных между собой графов с n вершинами и четырьмя ребрами, а затем подсчитать количество попарно неизоморфных графов.  
Попарная неизоморфность означает, пару графов которые не изоморфны между собой.  
Значит количество попарных неизоморфных графов означает количество максимально возможных таких пар.  
**Для проделанных ниже действий наглядное (иллюстрированное) объяснение находится в тестах.**Так как алгоритма подсчета максимально возможного количества неизоморфных графов не существует, придется считать это количество методом перебора.  
Для n=1 существует 1 граф, то есть количество попарно неизоморфных графов равно 0.  
Для n=2 существует 9 неизоморфных между собой графа, то есть количество попарно неизоморфных графов равно 8+7+6+5+4+3+2+1  
Для n=3 существует 27 неизоморфных между собой графа, то есть количество попарно неизоморфных графов равно 26+25+…+1.  
Для n=4 существует 45 неизоморфных между собой графа, то есть количество попарно неизоморфных графов равно 44+43+..+1.  
Для n=5 существует 58 неизоморфных между собой графов, то есть количество попарно неизоморфных графов равно 57+56+…+1.  
Для n=6 существует 65 неизоморфных между собой графов, то есть количество попарно неизоморфных графов равно 64+63+…+1.  
Для n=7 существует 67 неизоморфных между собой графов, то есть количество попарно не изоморфных графов равно 66+65+…+1.  
Для n=8 существует 68 неизоморфных между собой графов, то есть количество попарно не изоморфных графов равно 67+66+…+1.  
Заметим, что если пробовать строить графы с количеством вершин n>=9, то во всех случаях будет существовать всего 68 неизоморфных между собой графов, то есть количество попарно не изоморфных графов равно 9+8+7+6+5+4+3+2+1 для n>8.

Выделение основных подзадач

1. Счет количества попарно неизоморфных графов;

# 3. Структуры данных, используемые для представления исходных данных и результатов задачи

## Внешнее представление входных данных

Вещественные числа, которые находятся в файле input.txt.

## Внешнее представление выходных данных

Вещественные числа, которые находятся в файле output.txt.

# Внутреннее представление входных данных

Нециклический линейный однонаправленный список a. Без фиктивного звена. Каждое звено имеет 2 поля. 1 поле – вещественное число chislo, 2 поле next – адрес следующего звена. Struct a{float chislo; a \*next} – структурный тип a, где его составляющие chislo-Вещественное число. \*next-указатель.

## Внутреннее представление данных

\*L, \*curr – указатели структурного типа a. L-указывает на начало списка. Curr используется для обозначения текущего элемента.  
float t – вещественная переменная, хранящая значение элемента следующего после текущего.  
bool s – переменная типа bool хранящая результат true в случае если последовательность упорядочена, и false в случае если последовательность не упорядочена.  
FILE \*fp – указатель на поток файлов.  
\*sled, \*pred – указатели структурного типа a, где sled-следующий элемент, pred-предыдущий элемент. Область действия - подзадача menyaem().

Результат представлен в виде последовательности, где элементы этой последовательности – вещественные переменные. Количество элементов последовательности - n, где n – количество исходных вещественных чисел.   
Минимальное количество элементов последовательности - 2.

# 4. Укрупненный алгоритм решения задачи

Menyaem()

Начало

S=true;  
L = new a;

curr = L;  
t=curr->next

Curr=L; pred=NULL;

Sled=curr->next

curr->next = новый элемент

(curr->next) = pred;

pred = curr;

curr = sled;

sled = (sled->next);

Curr->chislo>t

Файл закончился?

S=false

sled->next = curr;

curr->next = pred;

L = sled;

curr = (curr->next);

curr->chislo = t;

Конец

Файл закончился?

curr->next = NULL;

S=false

Конец

Menyaem()

Вывод А

Этот алгоритм прост в понимании и не требует удаления лишних элементов так как их нет. Алгоритм действует по принципу: поменять направление списка, в этом заключается его легкость.

# Структура программы

## Взаимосвязь функций

menyaem()

Void main()

## Составные части программы

Наименование функции:   
menyaem  
Прототип функции:  
void menyaem();  
Данная подпрограмма инвертирует список, начало которого в L.

# Текст программы

# Набор тестов

Тест 1.  
Назначение: количество вершин n=1||2||3.  
Входные данные:   
3  
Результат:  
0  
Замечание: очевидно нельзя иллюстрировать простой неориентированный граф с 3 вершинами, так как такового не существует.  
Тест 2.  
Назначение: количество вершин n=4.  
Входные данные:  
4  
Результат:  
2  
Замечание:   
Всего 2 возможных неизоморфных между собой графа.

1. 2)

Тест 3.  
Назначение: количество вершин n=5.  
Входные данные:  
5  
Результат:  
15  
Замечание:  
Всего 6 возможных неизоморфных между собой графа.   
  
 1) 2) 3) 4)

5) 6)

Таким образом количество попарно неизоморфных графов= (1,2) +(1,3) +(1,4) +(1,5) +(1,6) +(2,3) +(2,4) +(2,5) +(2,6) +(3,4) +(3,5) +(3,6) +(4,5) +(4,6) + (5,6) =5+4+3+2+1=15.

Тест 6.  
Назначение: количество вершин n=8, максимальное количество неизоморфных между собой графов с 4 ребрами.  
Входные данные:  
8  
Результат:  
45  
Замечание:  
Всего 10 возможных неизоморфных между собой графа.

1. 2) 3) 4)

5) 6) 7) 8)

9) 10)

Таким образом количество попарно неизоморфных графов=(1,2)+(1,3)+...+(1,10)+(2,3)+(2,4)+…+(2,10)+(3,4)+…+(9,10)=9+8+7+6+5+4+3+2+1=45

# 8. Результат работы программы

-Программа выдала верное решение на всех тестах и, следовательно, правильно работает.