ФГОУ ВПО Кубанский государственный технологический университет

(«КубГТУ»)

Кафедра Информационных систем и программирования

Факультет Компьютерных технологий и автоматизированных систем

**КУРСОВАЯ РАБОТА**

По дисциплине Дискретная математика

На тему Задачи и алгоритмы дискретной математики

Выполнил(а) студент(ка) группы 12-КБ-ПР1

Межераунис Ангелина Константиновна

Допущен(а) к защите

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Защищён \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Симоненко Е. А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Волик А. Г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Краснодар 2013

ФГОУ ВПО Кубанский государственный технологический университет

(«КубГТУ»)

Кафедра Информационных систем и программирования

Факультет Компьютерных технологий и автоматизированных систем

УТВЕРЖДАЮ

Зав. кафедрой ИСП

проф., д.т.н. Видовский Л.А.

(дата, подпись, расшифровка подписи)

**ЗАДАНИЕ**

на курсовую работу

Студентке Межераунис Ангелине Константиновне группы 12-КБ-ПР1

факультета Компьютерных технологий и автоматизированных систем

специальности 231000 – Программная инженерия

Тема работы Задачи и алгоритмы дискретной математики.

Содержание задания: Изучить темы «Числа Каталана. Вычисление и применение» и «Рёберная раскраска. Алгоритм рёберной раскраски», провести исследования алгоритмов работы с этими объектами, составить программы, которые демонстрируют указанные алгоритмы, провести тестирование программ.

Объём курсовой работы:

а) пояснительная записка стр.16;

б) графическая часть лист формата А4;

Рекомендуемая литература Седжвик «Алгоритмы на С++», Скиена «Алгоритмы», Окулов «Дискретная математика».

Срок выполнения проекта: с 2 сентября 2013г. по 21 декабря 2013г.

Срок защиты: со 2 сентября 2013г. по 21 декабря 2013г.

Дата выдачи задания: с 3 сентября 2013г. по 10 декабря 2013г.

Дата сдачи работы на кафедру: с 25 сентября 2013г. по 28 декабря 2013г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

РЕФЕРАТ

ЧИСЛА КАТАЛАНА, ПОСЛЕДОВАТЕЛЬНОСТЬ, РЕКУРСИЯ, РАСКРАСКА РЁБЕР, ГРАФ.

Стр.16, рис. 8, библ. 3 .

В данной курсовой работе были рассмотрены числа Каталана, их взаимосвязь с различными конструкциями, метод генерации последовательности чисел с помощью рекуррентной формулы, нахождение числа по номеру с помощью не рекуррентной формулы, а также алгоритм рёберной раскраски, его применение и реализация. Основным моментам исследования чисел Каталана было раскрытие взаимосвязи между различными конструкциями (скобочные последовательности, бинарные деревья и т.д.) через последовательность чисел. Во второй части наиболее важным было понимание способов практического применения алгоритма раскраски рёбер и его реализация на языке С++.

Следовательно, данная работа сделала возможным осознание важности изучения основных понятий дискретной математики.

Содержание

Введение………………………………………………………………………………………….5

[Глава 1 Числа Каталана 5](#_Toc371092865)

1.1. Формулировка задачи……………………………………………………………………….5

1.2. Результаты тестирования…………………………………………………………………...9

[Глава 2 Раскраска рёбер 11](#_Toc371092866)

2.1. Формулировка задачи……………………………………………………………………...11

2.2. Результаты тестирования………………………………………………………………….12

Список используемых источников………………...…………………………………………..13

Приложение А…………………………………………………………………………………..14

Приложение Б…………………………………………………………………………………..15

Приложение В…………………………………………………………………………………..16

# Введение

Дискретная математика является одной из самых необходимых дисциплин для программиста. Этот предмет изучает различные задачи, глубинное понимание которых помогает в разработке наиболее оптимальных алгоритмов. Главное разобраться, каким образом применять на практике полученную информацию.

Числа Каталана – удивительная последовательность. Она замечательна тем, что связывает далёкие друг от друга конструкции и позволяет решать сложные задачи, например, с деревьями, или маршрутами в четырёх угольнике через скобочные последовательности. То есть с помощью этих чисел можно находить довольно простые и оригинальные способы решения задач.

Также алгоритм раскраски рёбер крайне полезен в задачах, связанных с расписанием. Например, его можно использовать для планировки спортивных состязаний. А также, видоизменяя алгоритм, можно использовать его в различных задачах, связанных с распределением времени, при некоторых дополнительных условиях.

Таким образом, в данной работе рассмотрены полезные на практике алгоритмы, что в частном случае подтверждает важность такой математической дисциплины, как дискретная математика

# Глава 1. Числа Каталана

* 1. **Формулировка задачи:**

Изучить тему «Числа Каталана. Вычисление и применение», исследовать алгоритм генерации последовательности чисел, способы применения чисел, составить программу, демонстрирующую работу алгоритма, провести ручное и машинное тестирование.

Генерация последовательности чисел.

Число Каталана выражается формулой C(n) = (2n)!/n!(n+1)!

Начало последовательности выглядит так: 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796…

Кроме того числа Каталана можно вычислить с помощью рекуррентной формулы.

Сn = С0Cn-1 + С1Cn-2 + С2Cn-3 +…+Сn-1 + Cn-1C0 =

Применение чисел Каталана.

Известно, как минимум 66 различных конструкций, к которым можно применить последовательность чисел Каталана. Некоторые из них были рассмотрены в данной работе.

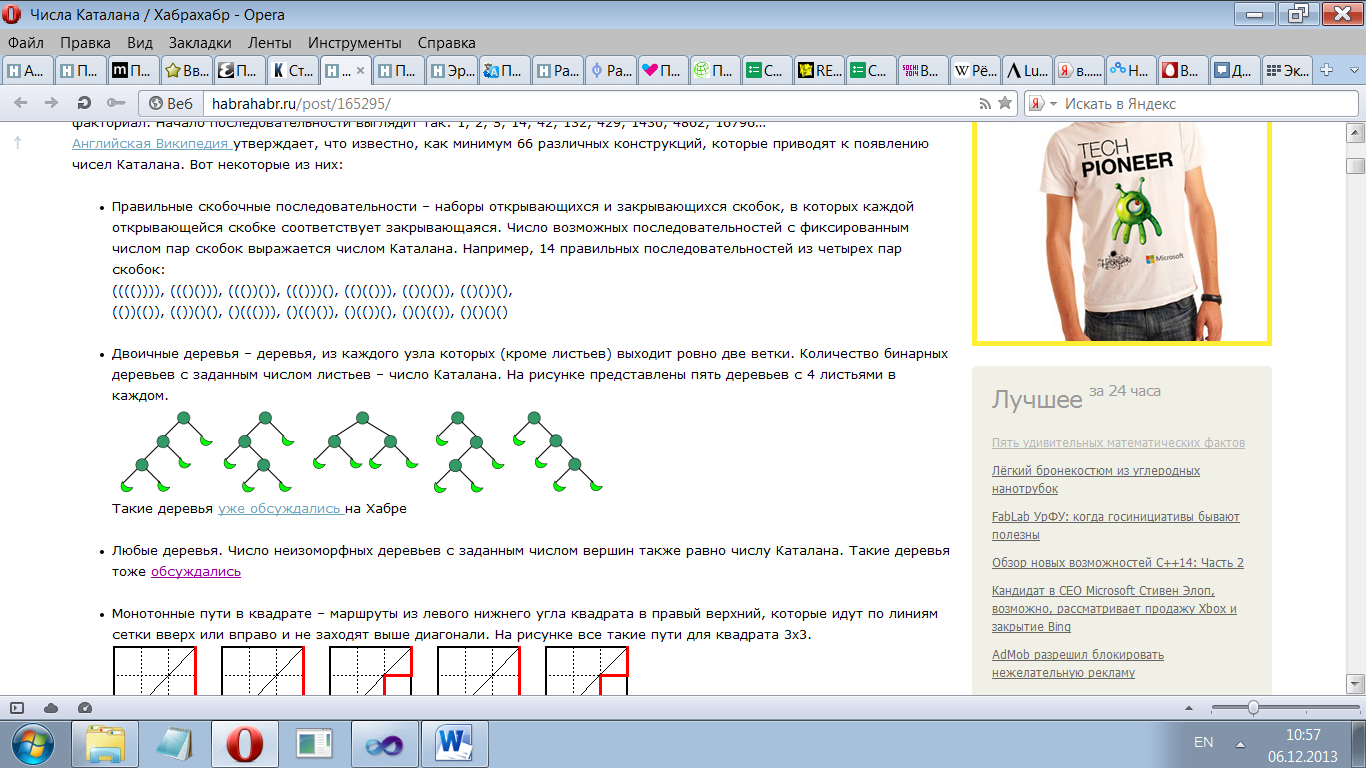
1.Правильные скобочные последовательности – наборы открывающихся и закрывающихся скобок, в которых каждой открывающейся скобке соответствует закрывающаяся. Число возможных последовательностей с фиксированным числом пар скобок выражается числом Каталана.

Пример для третьего числа.

((())), (()()), ()()(), ()(()), (())()

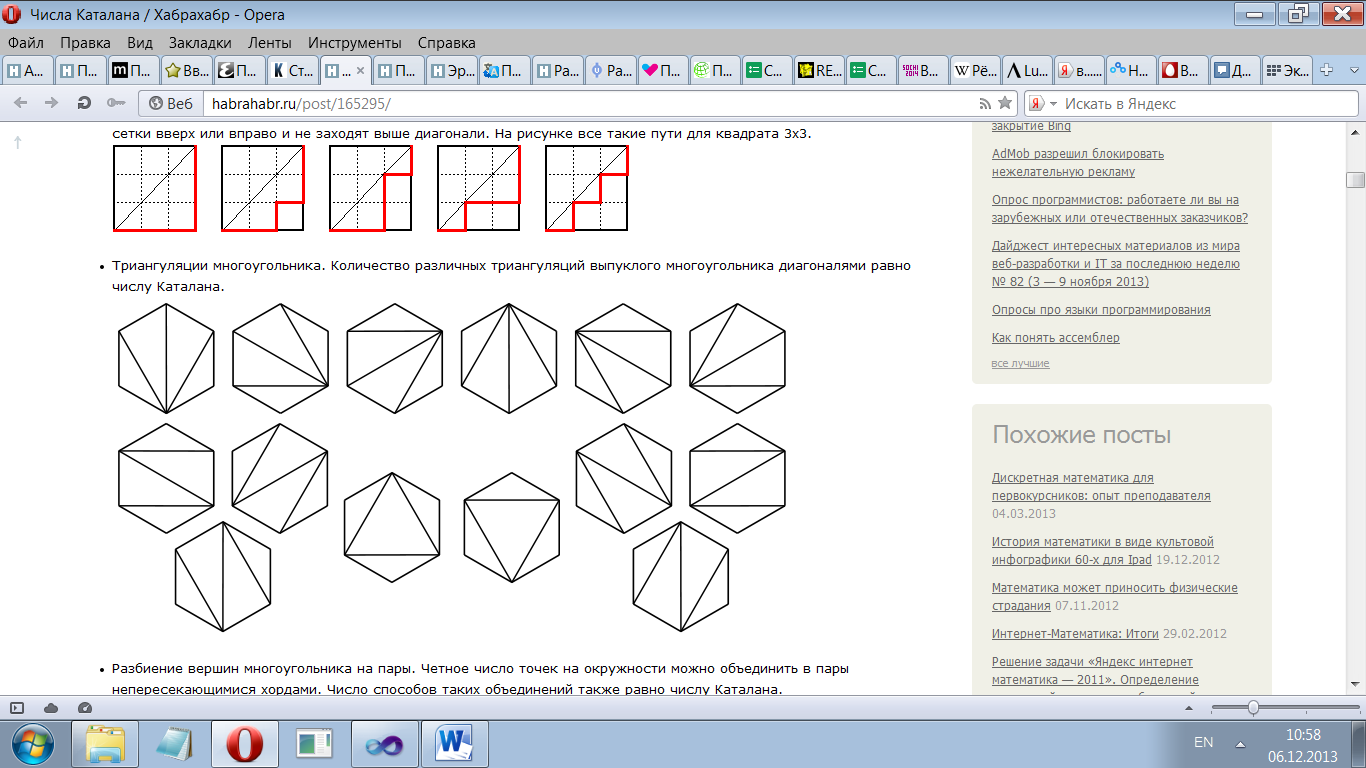
2. Двоичные деревья – деревья, из каждого узла которых (кроме листьев) выходит ровно две ветки. Число узлов дерева соответствует номеру числа Каталана в последовательности. Количество возможных бинарных деревьев с данным количеством узлов и есть число Каталана.

Пример для третьего числа.



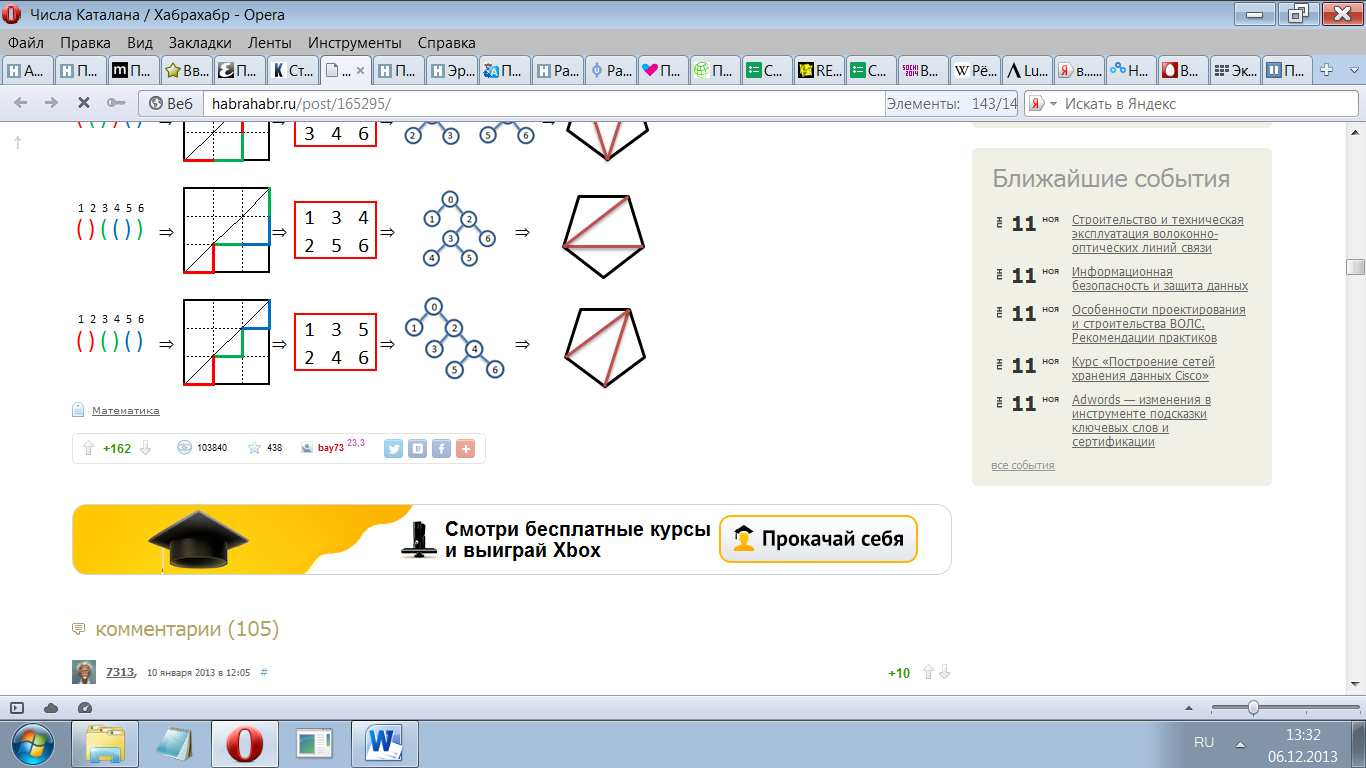
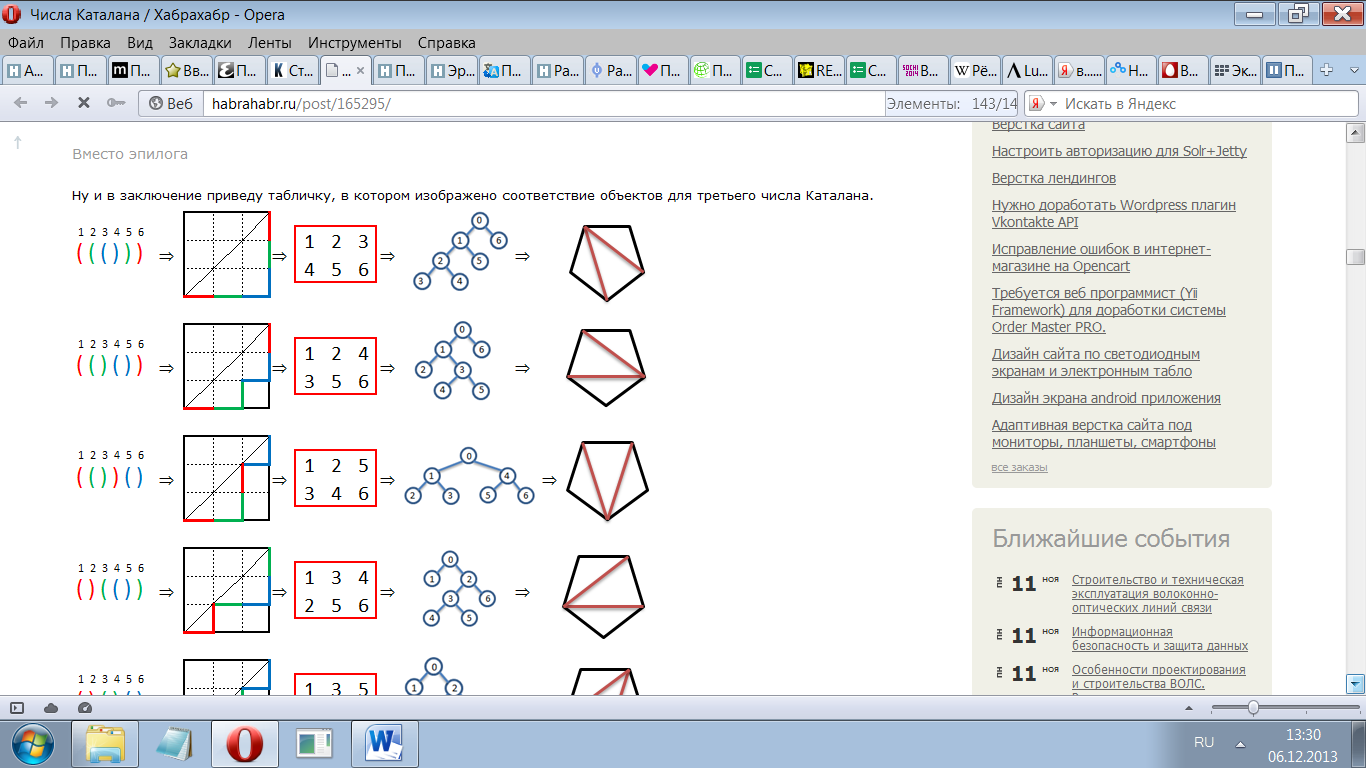
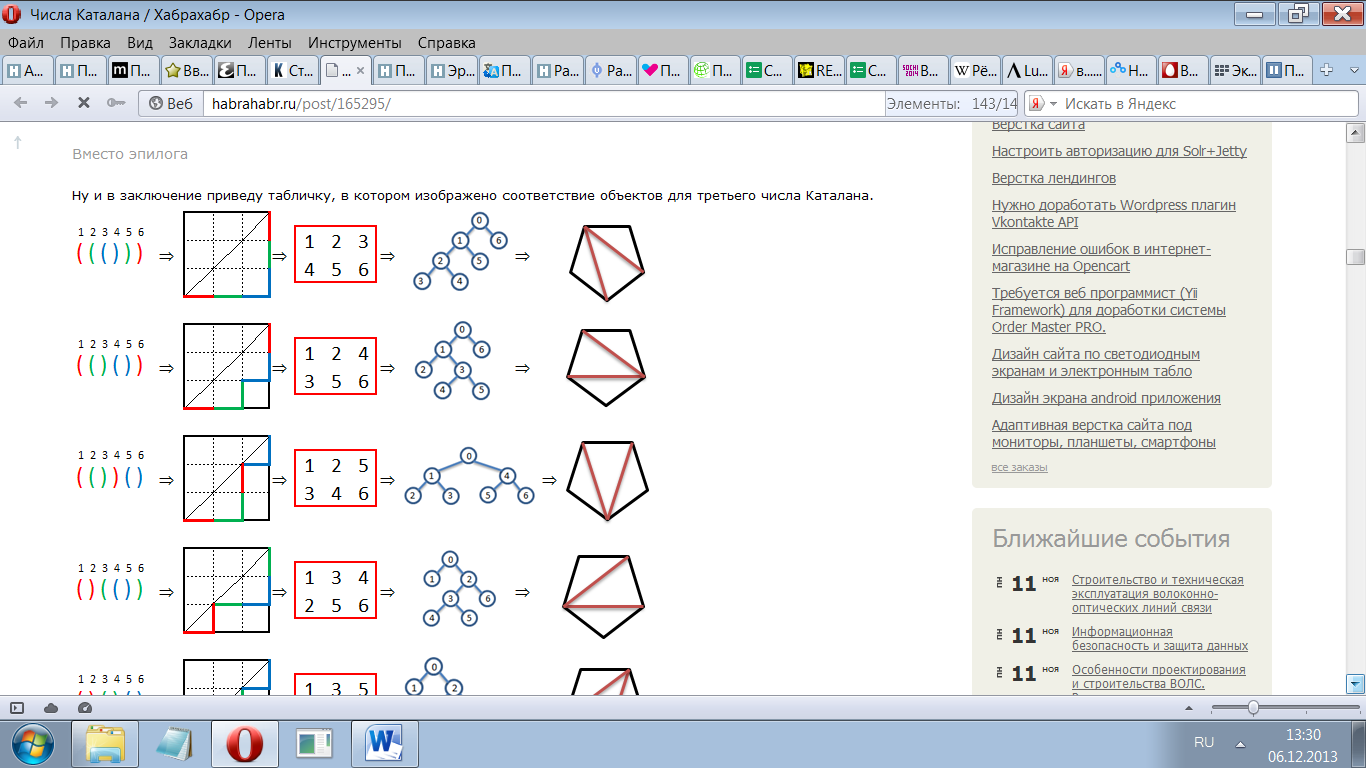
3. Монотонные пути в квадрате – маршруты из левого нижнего угла квадрата в правый верхний, которые идут по линиям сетки вверх или вправо и не заходят выше диагонали.

Пример для третьего числа.



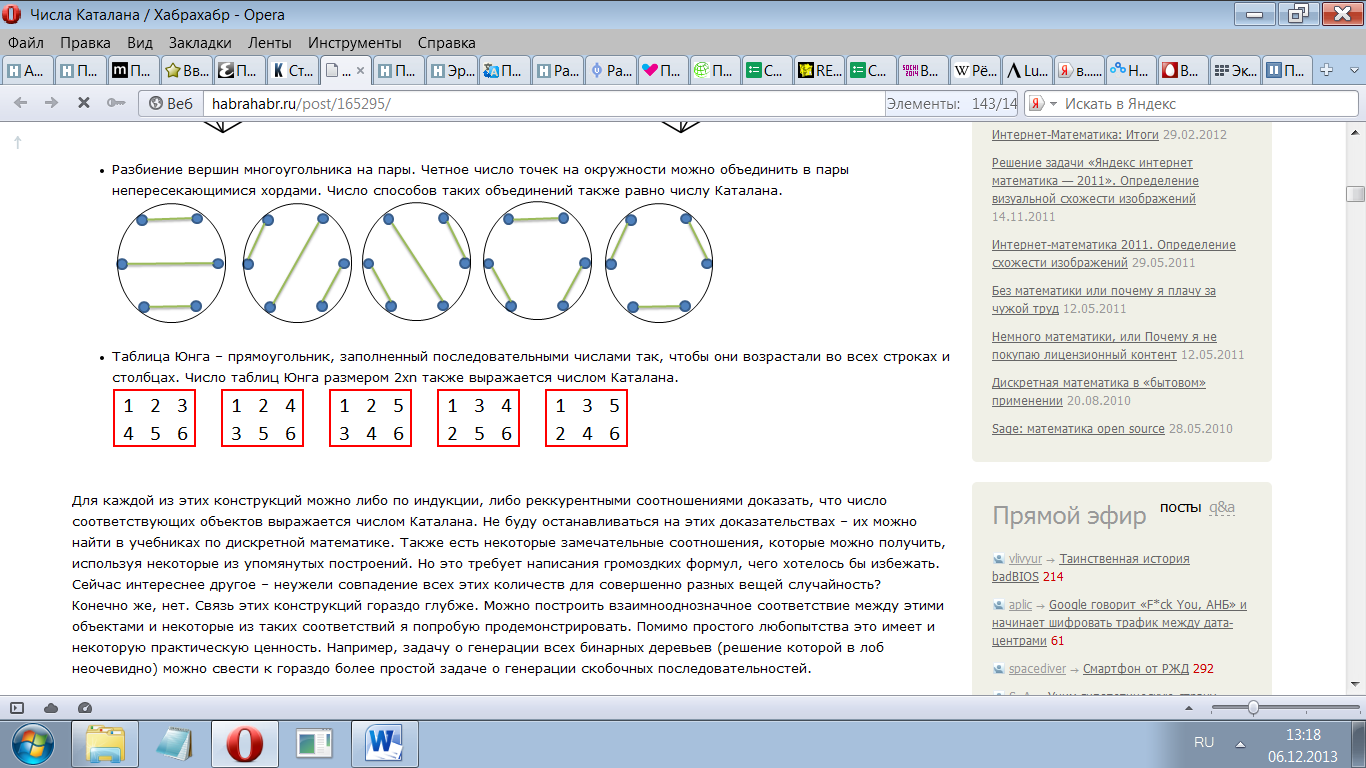
4.Триангуляции многоугольника. Количество различных триангуляций выпуклого многоугольника диагоналями соответствуют числу Каталана с номером равным числу диагоналей плюс один.

Пример для третьего числа.



5. Таблица Юнга – прямоугольник, заполненный последовательными числами так, чтобы они возрастали во всех строках и столбцах. Число таблиц Юнга размером 2xn также выражается числом Каталана.

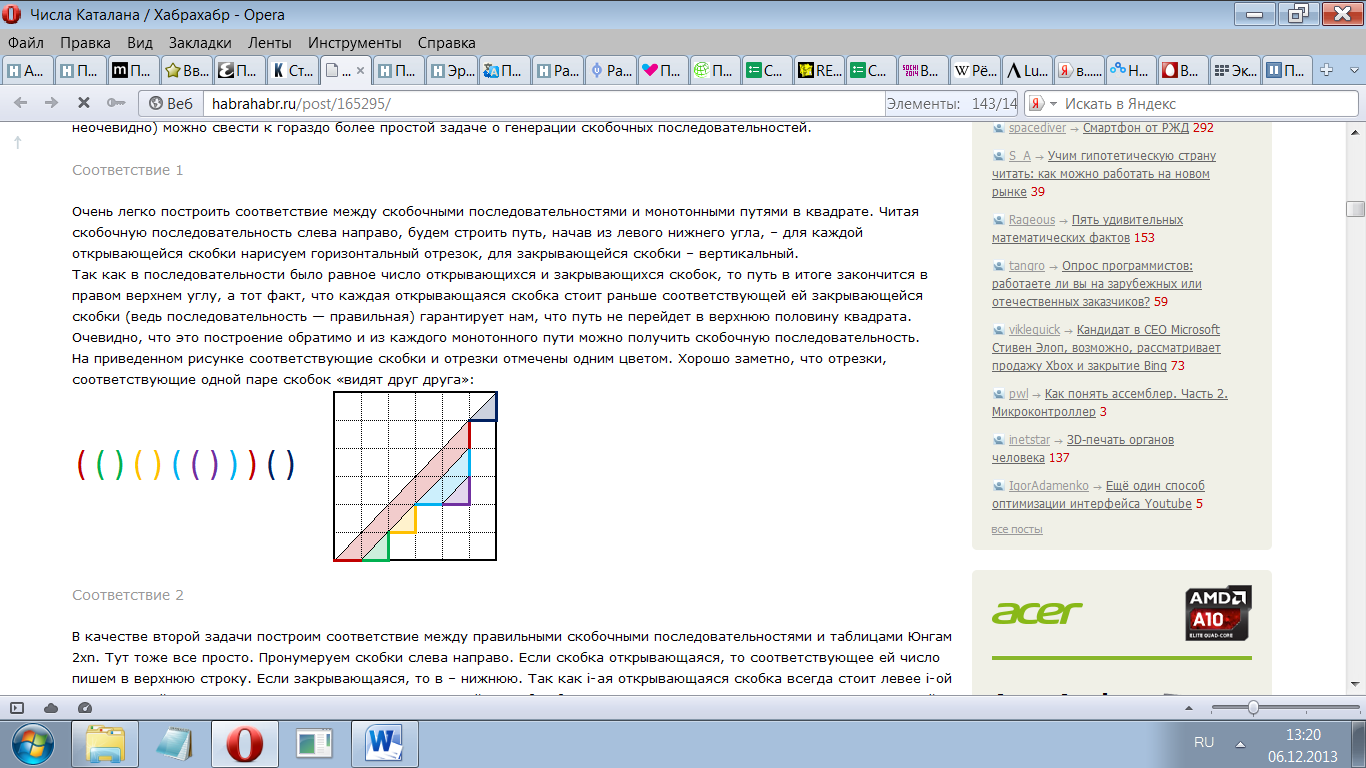
Пример для третьего числа.



Между всеми этими объектами существует взаимно-однозначное соответствие. Некоторые из них освещены в работе. Эта взаимосвязь через числа Каталана имеет определённую практическую ценность. Например, довольно сложную задачу о генерации всех бинарных деревьев можно свести к гораздо более простой задаче о генерации скобочных последовательностей.

1.Соответствие между скобочными последовательностями и монотонными путями в квадрате. Строится оно следующим образом: читая скобочную последовательность слева направо, будем строить путь, начав из левого нижнего угла, – для каждой открывающейся скобки нарисуем горизонтальный отрезок, для закрывающейся скобки – вертикальный.

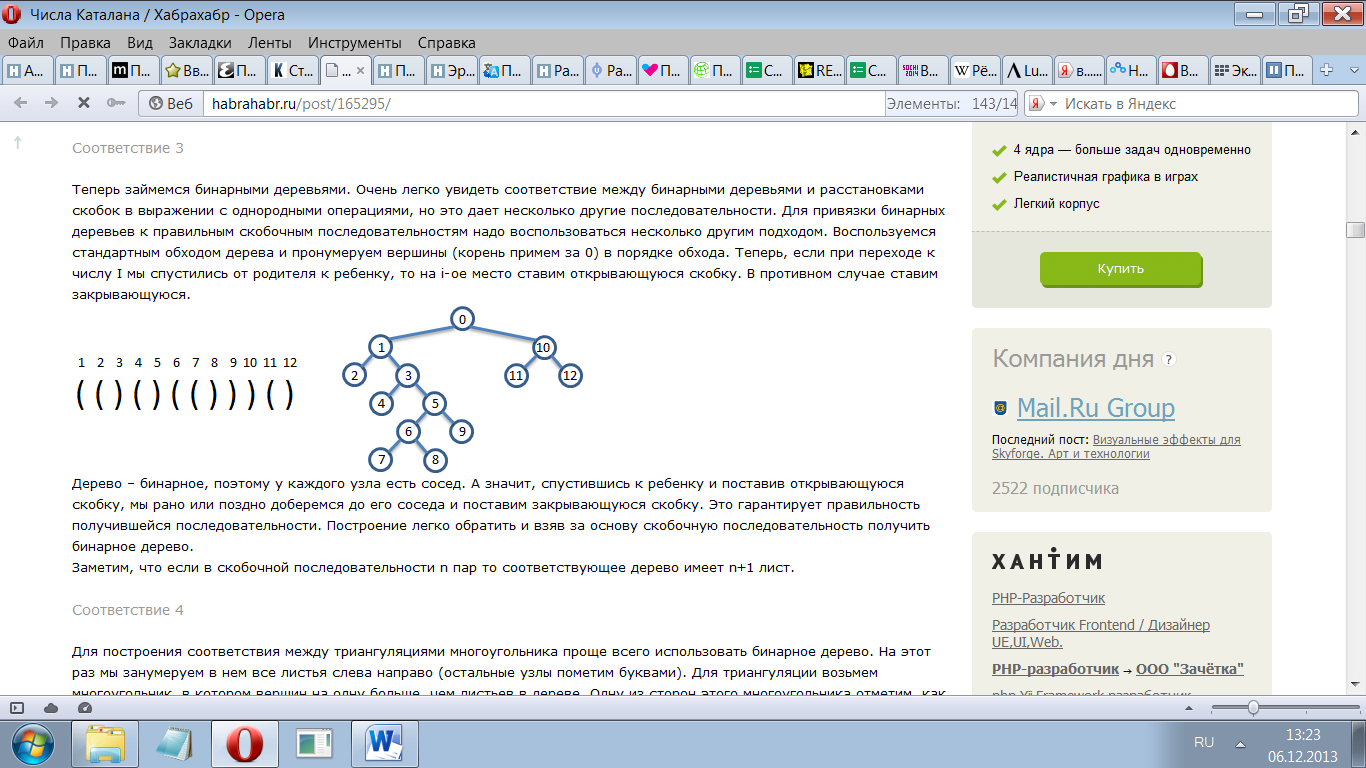
Так как в последовательности было равное число открывающихся и закрывающихся скобок, то путь в итоге закончится в правом верхнем углу, а тот факт, что каждая открывающаяся скобка стоит раньше соответствующей ей закрывающейся скобки (ведь последовательность — правильная) гарантирует нам, что путь не перейдет в верхнюю половину квадрата. Очевидно, что это построение обратимо и из каждого монотонного пути можно получить скобочную последовательность.



2. Соответствие между правильными скобочными последовательностями и таблицами Юнгам 2xn. Пронумеруем скобки слева направо. Если скобка открывающаяся, то соответствующее ей число пишем в верхнюю строку. Если закрывающаяся, то в – нижнюю. Так как i-ая открывающаяся скобка всегда стоит левее i-ой закрывающейся, то число соответствующее открывающейся скобке будет меньше числа, соответствующего закрывающей. А значит, верхнее число в таблице окажется меньше нижнего в той же колонке, то есть из правильной скобочной последовательности мы получили таблицу Юнга. Это построение также обратимо, а значит получено взаимно-однозначное соответствие.

3. Соответствие между бинарными деревьями и расстановками скобок.

Для привязки бинарных деревьев к правильным скобочным последовательностям надо воспользоваться несколько другим подходом. Воспользуемся стандартным обходом дерева и пронумеруем вершины (корень примем за 0) в порядке обхода. Теперь, если при переходе к числу I мы спустились от родителя к ребенку, то на i-ое место ставим открывающуюся скобку. В противном случае ставим закрывающуюся. Дерево – бинарное, поэтому у каждого узла есть сосед. А значит, спустившись к ребенку и поставив открывающуюся скобку, мы рано или поздно доберемся до его соседа и поставим закрывающуюся скобку. Это гарантирует правильность получившейся последовательности. Построение легко обратить и взяв за основу скобочную последовательность, получить бинарное дерево.



Заметим, что если в скобочной последовательности n пар, то соответствующее дерево имеет n узлов.

4. Соответствия между триангуляциями многоугольника бинарными деревьями.

На этот раз мы занумеруем в нем все листья слева направо (остальные узлы пометим буквами). Для триангуляции возьмем многоугольник, в котором вершин на одну больше, чем листьев в дереве. Одну из сторон этого многоугольника отметим, как стартовую, а остальные занумеруем (для наглядности – против часовой стрелки).

Далее выполняем следующую процедуру – если две вершины дерева соседние, то соответствующие стороны многоугольника «стянем» диагональю, которую пометим той буквой, которой помечен родитель этой пары узлов в дереве. Далее продолжаем процедуру «стягивания» пока от многоугольника не останется единственный стартовый отрезок.

Как можно заметить три стороны каждого треугольника в получившемся разбиении соответствуют одному родительскому узлу и двум его потомкам. Поэтому, если взять два разных дерева, то получится два разных разбиения.

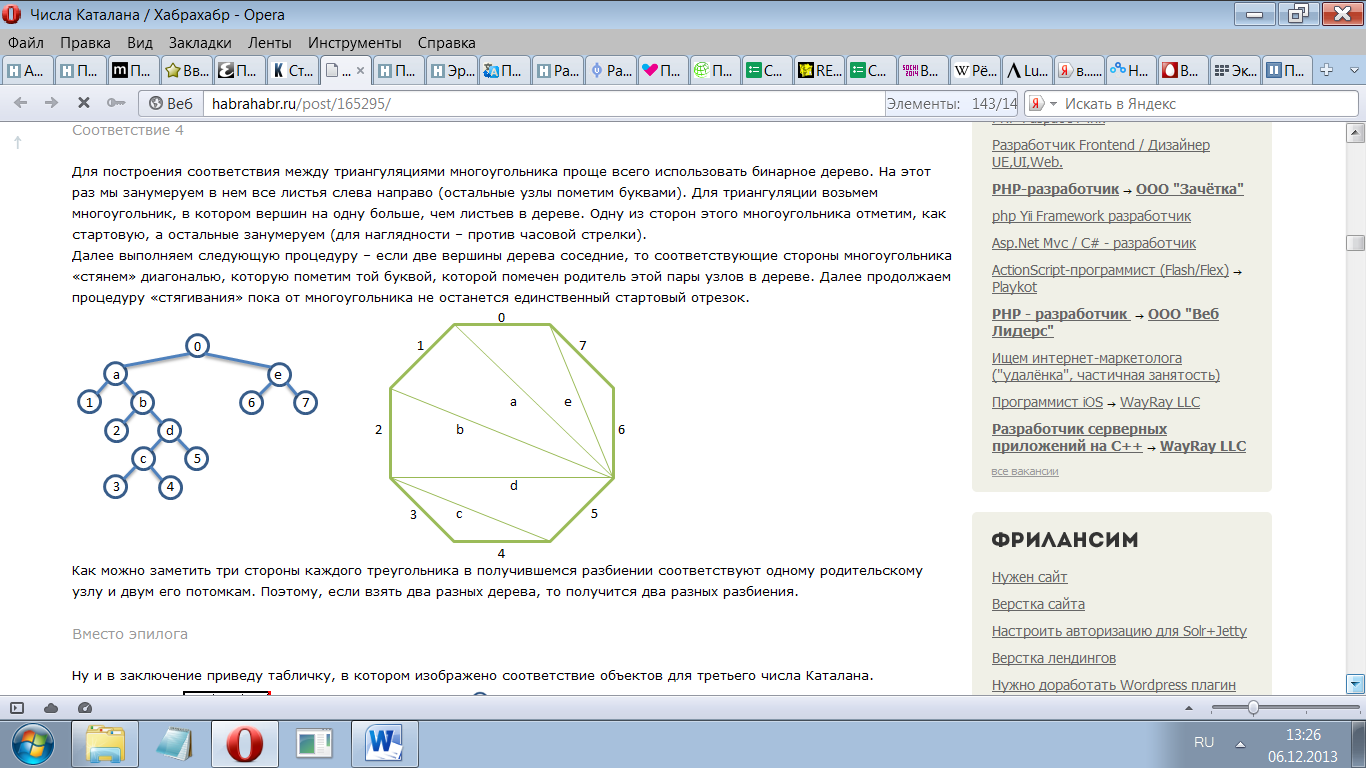
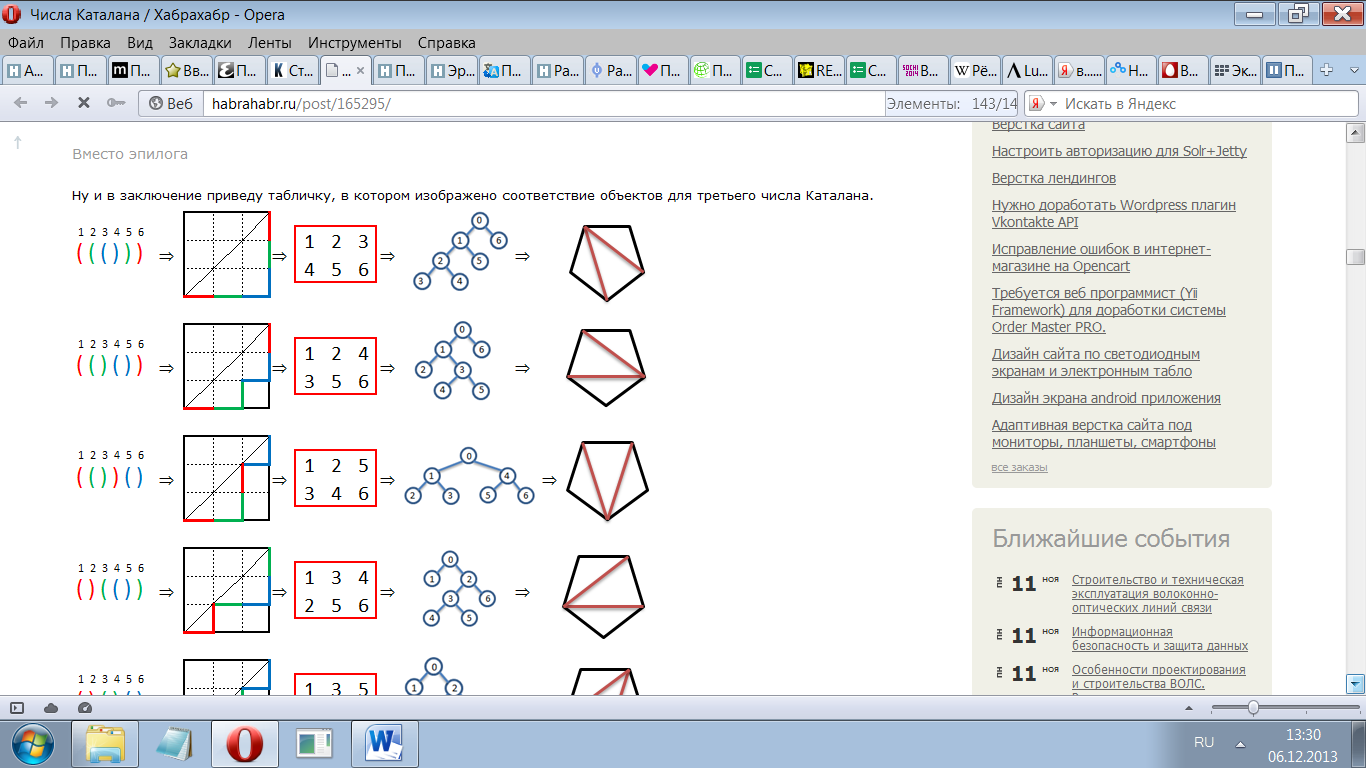


Таблица соответствий для третьего числа Каталана. 

Результаты тестирования программы, выполняющей генерацию последовательности чисел Каталана:

На вход подаётся n – количество чисел, которые необходимо сгенерировать.

На выход – последовательность из n чисел.

Вход: 15

Выход:

1

2

5

14

42

132

429

1430

4862

16796

58786

208012

742900

2674440

9694845

# 1.2. Результаты тестирования

На вход подаётся n- номер числа в последовательности.

На выход – число Каталана под заданным номером.

1)Вход: 4

Выход: 14

2)Вход:6

Выход:132

3)Вход: 10

Выход: 16796

4)Вход:18

Выход: 2674440

# Глава 2. Раскраска рёбер

**2.1. Формулировка задачи:**

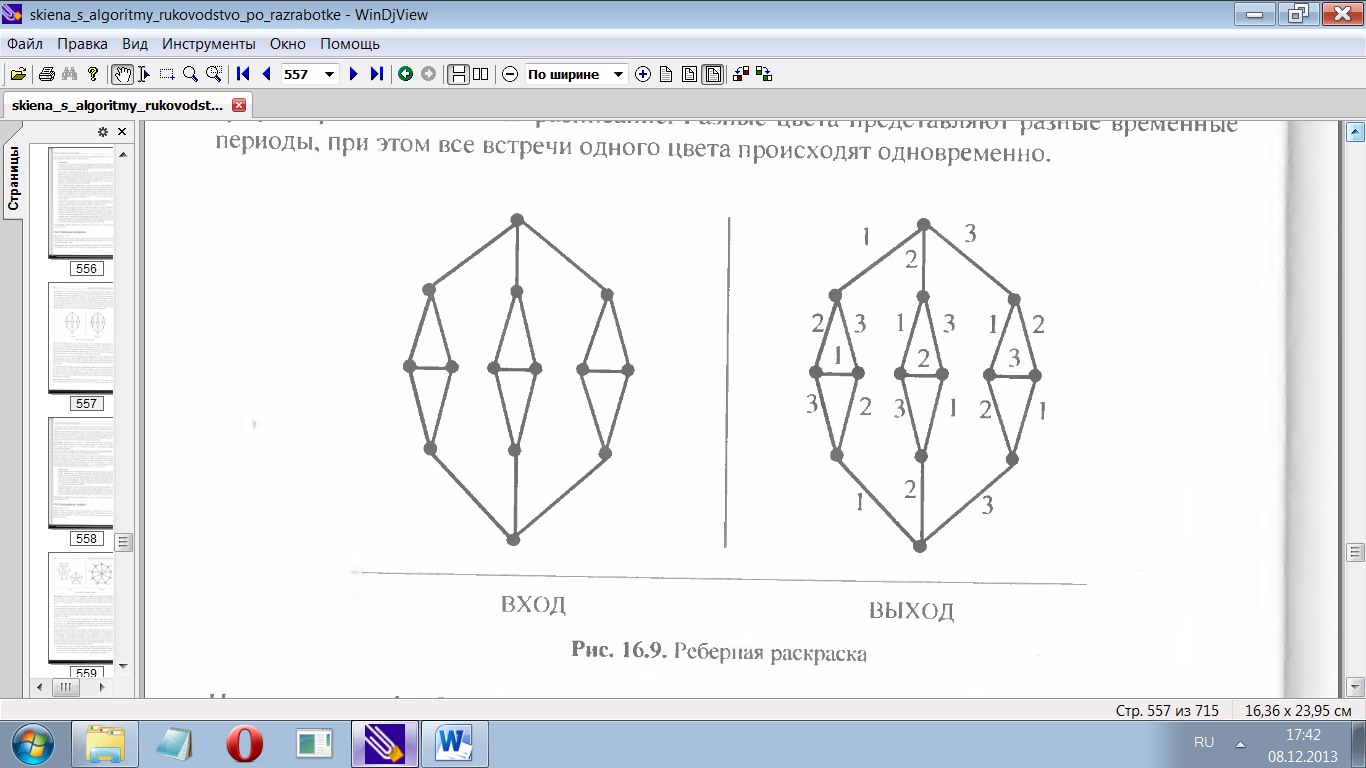
Изучить тему «Раскраска рёбер», исследовать алгоритм раскраски рёбер графа, составить программу, демонстрирующую работу алгоритма, провести ручное и машинное тестирование.

Задача рёберной раскраски.

Найти наименьший набор цветов, требуемый для раскраски рёбер некоторого графа таким образом, чтобы не одна пара рёбер, имеющих общую вершину, не была окрашена одним цветом.

Задачи рёберной раскраски имеют приложения в задачах календарного планирования, расписания. В общем, где необходимо минимизировать количество взаимно конфликтующих маршрутов.

Для примера рассмотрим ситуацию, где нужно составить расписание нескольких деловых встреч длиной один час, в каждой из которых участвуют два сотрудника. Чтобы избежать конфликтов, можно просто запланировать все встречи на разное время, но разумнее будет назначить неконфликтующие встречи на одно и то же время. Для решения этой задачи создадим граф, в котором вершины представляют людей, а рёбра задают тех, которые должны встретиться. Рёберная раскраска этого графа определяет искомое расписание. Разные цвета представляют разные временные периоды, при этом все встречи одного цвета проходят в одно время.



Минимальное число цветов, в которые мы можем раскрасить граф, называется хроматическим индексом. В данном примере он равен трём. То есть в этом случае требуется минимум три часа, чтобы состоялись встречи между тринадцатью людьми.

Согласно теореме Визинга наиболее распростронены 2 типа графов. У первого типа хроматическое число равно максимальной степени вершины, и второй – максимальной степени вершины плюс один.

Самый быстрый из известных алгоритмов для нахождения хроматического числа работает за экспоненциальное время.

**2.2. Результаты тестирования:**

На вход подаётся количество вершин в графе и матрица инцидентности.

На выход – хроматическое число графа первого типа.

1)Вход:

4

0 1 1 1

1 0 0 0

1 0 0 1

1 0 1 0

Выход:

3

2)Вход:

7

0 1 1 1 0 0 0

1 0 0 0 1 0 0

1 0 0 1 0 1 0

1 0 1 0 1 0 1

0 1 0 1 0 0 0

0 0 1 0 0 0 1

0 0 0 1 0 1 0

Выход:

3

# Список используемых источников

1. Скиена С. Алгоритмы. Руководство по разработке. – 2-е изд.: пер. с англ. – СПб.: БХВ-Петербург, 2011. – 720 с. Скиена «Алгоритмы».
2. Седжвик Р. Алгоритмы на C++. – Пер. с англ. – М.: Вильямс, 2011. – 1056 с.
3. Окулов С.М. Дискретная математика. Теория и практика решения задач по информатике: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2008. – 422 с.
4. Кристофидес Н. Теория графов. Алгоритмический подход: учебное пособие. – М.: Мир, 2008. – 427 с.

Приложение А

Код программы, генерирующей последовательность n чисел Каталана с помощью рекуррентной формулы.

#include <iostream>

using namespace std;

int main()

{

int a[100];//массив для хранения последовательности

a[0] = 1;//С0=1

int n; //количество чисел в последовательности

cin >> n;

for(int i=1; i <= n; i++)

{

a[i] = 0;//текущее значение, считаемого числа 0

for(int j=0; j<i; j++)

{

a[i] = a[i] + a[0+j]\*a[i-1-j];//считаем по формуле

}

}

for (int i=1; i<=n; i++)

{

cout << a[i] << " ";

}

return 0;

}

**Приложение Б**

Листинг программы, генерирующей последовательность n чисел Каталана с помощью нерекуррентной формулы.

#include <iostream>

using namespace std;

long long funk ( long long k)//функция вычисляющая факториал

{

if (k!=1)

{

k = k\*funk(k-1);

}

else

return k;

}

int main ()

{

long long n;//номер числа в последовательности

cin >> n;

cout << funk(2\*n)/(funk(n+1)\*funk(n));

return 0;

}

**Приложение В**

#include <iostream>

using namespace std;

int main ()

{

int n;//количество вершин

int a[100][100];//матрица ицидентности

for (int i=0; i<n; i++)

for (int j=0; j<n; j++)

{

cin >> a[i][j];

}

int k[100]; //массив,в кот. будем сохранять степени вершин

for (int i=0; i<n; i++)

k[i]=0;

for (int i=0; i<n; i++)

for (int j=0; j<n; j++)

{

k[i]=k[i] + a[i][j];

}

int max=0; //максимальная степень вершины

for (int i=0; i<n; i++)

{if(k[i] > max)

{

max = k[i];

}

}

cout << max;

return 0;

}