МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Алгоритмы на планарных графах

АВТОРЕФЕРАТ

дипломной работы

студентки 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Титковой Анастасии Дмитриевны

Научный руководитель

д. ф.-м. н., доцент ___________ М. Б. Абросимов

21.01.2023 г.

Заведующий кафедрой

д. ф.-м. н., доцент ___________ М. Б. Абросимов

21.01.2023 г.

Саратов 2023

ВВЕДЕНИЕ

Во многих случаях не имеет значения как изобразить граф, поскольку изоморфные графы несут одну и ту же информацию. Но встречаются такие ситуации, когда необходимо выяснить, есть ли возможность нарисовать граф на плоскости так, чтобы его изображение удовлетворяло определенным требованиям. Например, при проектировании железнодорожных путей, печатных плат и различных схем нам нежелательны пересечения. Именно для таких целей нам и необходимы планарные графы.

Планарность графа можно определить по различным критериям и алгоритмам. В данной работе для проверки планарности графа мы рассмотрим Гамма-алгоритм. Также рассмотрим алгоритмы визуализации всех графов с использованием физических аналогий и визуализации планарных графов методом сдвига. Рассмотрим алгоритм генерации случайных планарных графов.

В практической части будет представлена программа, которая позволяет проверить граф на планарность и отрисовать его несколькими способами. Непланарые графы можно визуализировать по окружности и с использованием метода физических аналогий, а планарные еще и методом сдвига. В программе реализована функция генерации случайных планарных графов. Также программа позволяет определить какие графы в заданном файле являются планарными. В программе для проверки планарности реализован модифицированный Гамма-алгоритм, который позволяет работать с любыми графами. Дополнительно проведем анализ: сколько графов подходит под непосредственное применение Гамма-алгоритма в его изначальном виде.

Дипломная работа состоит из введения, 2 разделов, заключения, списка использованных источников и 4 приложений. Общий объем работы – 80 страниц, из них 40 страниц – основное содержание, включая 41 рисунок и 1 таблицу, список использованных источников из 17 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Теоретическая часть

1.1 Основные определения

Неориентированным графом (или, для краткости, *графом*) называется пара $G = (V, \alpha)$, где α – симметричное и антирефлексивное отношение на множестве вершин V.

Дуги неориентированного графа обычно называют *ребрами*.

Изображение графа G называется *плоским*, если никакие два ребра в этом изображении не пересекаются, кроме быть может вершин.

Графы, допускающие плоское изображение, называют *планарными*.

Пусть планарный граф G задан плоским изображением. *Гранями* в этом изображении называются области, ограниченные ребрами графа. Одной их них является *внешняя грань*, представляющая собой бесконечную по протяженности часть плоскости.

Плоский граф называется *триангуляцией*, если все его грани – треугольники.

Пусть построена некоторая укладка подграфа $\tilde{G} = (\tilde{V}, \tilde{\alpha})$ графа $G = (V, \alpha)$. *Сегментом* S *относительно* $\tilde{G}$ (иногда просто *сегментом*) будем называть подграф графа G одного из следующих двух видов:

1. Ребро $\{u, v\} \in \alpha$ такое, что $\{u, v\} \notin \tilde{\alpha}$, $u, v \in \tilde{V}$;

2. Связную компоненту графа $G - \tilde{G}$, дополненную всеми ребрами графа G, инцидентными вершинами взятой компоненты, и концами этих ребер.

Вершину v сегмента S относительно $\tilde{G}$ будем называть *контактной*, если $v \in \tilde{V}$.

Поскольку граф $\tilde{G}$ плоский, то он разбивает плоскость на грани. *Допустимой гранью для сегмента* S относительно $\tilde{G}$ называется грань Γ графа $\tilde{G}$, содержащая все контактные вершины сегмента S. Через $\Gamma(S)$ будем обозначать множество допустимых граней для S. Может оказаться, что $\Gamma(S) = \emptyset$.

Простую цепь L сегмента S, соединяющую две различные контактные вершины и не содержащую других контактных вершин, назовем α-*цепью*. Очевидно, что всякая α-цепь, принадлежащая сегменту, может быть уложена в любую грань, допустимую для этого сегмента.

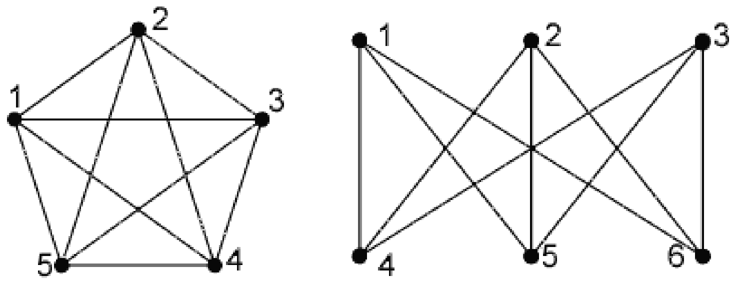
1.2 Планарные графы

Теорема 1 (формула Эйлера).

Для плоского изображения связного планарного графа G, имеющего n вершин и m ребер, справедливо равенство $n - m + r = 2$, где r – количество граней (включая внешнюю).

Теорема 2 (теорема Понтрягина-Куратовского).

Граф планарен тогда и только тогда, когда он не содержит подграфов, гомеоморфных K_5 (рисунок 1а) или $K_{3,3}$ (рисунок 1б).



а б

Рисунок 1 – Графы K_5 и $K_{3,3}$

1.3 Гамма-алгоритм

На вход алгоритму подаются графы со следующими свойствами:

1. Граф должен быть связным.

2. Граф должен содержать хотя бы один цикл.

3. Граф не должен иметь мостов.

Теперь перейдем к самому алгоритму.

0. Выберем некоторый простой цикл C графа G и уложим его на плоскости; положим $\tilde{G} = C$.

1. Найдем грани графа $\tilde{G}$ и сегменты относительно $\tilde{G}$. Если множество сегментов пусто, то перейдем к пункту 7.

2. Для каждого сегмента S определим множество $\Gamma(S)$.

3. Если существует сегмент S, для которого $\Gamma(S) = \emptyset$, то граф G непланарен. Конец. Иначе перейдем к пункту 4.

4. Если существует сегмент S, для которого имеется единственная допустимая грань Γ, то перейдем к пункту 6. Иначе – к пункту 5.

5. Для некоторого сегмента S $(\Gamma(S) > 1)$ выбираем произвольную допустимую грань Γ.

6. Поместим произвольную α-цепь $L \in S$ в грань Γ; заменим $\tilde{G}$ на $\tilde{G} \cup L$ и перейдем к пункту 1.

7. Построена укладка $\tilde{G}$ графа G на плоскости. Конец.

1.4 Визуализация графов методом физических аналогий

При нахождении изображения графа можно рассматривать граф как систему тел с силами, взаимодействующими между телами, например, считая вершины графа телами, а ребра пружинами.

Наиболее простой подход состоит в использовании комбинации пружин и электронных сил, когда каждое ребро рассматривается как пружина, а вершины считаются одинаково заряженными частицами, между которыми действуют силы отталкивания. Более точно, сила, приложенная к вершине p, определяется по формуле (1):

$$F(p) = \sum_{u=(p,q)\in E} f_u + \sum_{(p,q)\in V^2} g_{(p,q)} \quad (1),$$

где f_u – сила растяжения, действующая на вершину p из-за пружины (p,q), а $g_{(p,q)}$ – это сила отталкивания, существующая между частицами p и q. По закону Фуке f_u пропорциональна разности между расстоянием от p до q и длиной пружины с минимальной энергией, а сила $g_{(p,q)}$ следует обратному квадратичному закону.

Пусть $d_{(p,q)}$ обозначает расстояние на плоскости между p и q. Тогда для x-й координаты силы $F(p)$ можно использовать формулу (2):

$$\sum_{u=(p,q)\in E} k_u^{(1)} \left(d(p,q) - l_u\right) \frac{x_p - x_q}{d(p,q)} + \sum_{(p,q)\in V^2} \frac{k_{(p,q)}^{(2)}}{(d(p,q))^2} \frac{x_p - x_q}{d(p,q)} \quad (2),$$

где l_u, $k_u^{(1)}$, $k_{(p,q)}^{(2)}$ – параметры, которые не зависят от позиции вершин на плоскости и интерпретируются следующим образом:

- l_u – это естественная (с нулевой энергией) длина пружины между p и q;
- $k_u^{(1)}$ – коэффициент жесткости (упругости) пружины между p и;
- $k_{(p,q)}^{(2)}$ – коэффициент силы отталкивания между p и q.

1.5 Визуализация планарных графов

Пусть G есть максимальный планарный граф с n вершинами, пусть u_0, u_1, u_2 являются *внешними вершинами* G (то есть лежащими на внешней грани) в порядке против часовой стрелки. *Каноническим порядком* G назовем такой порядок вершин v_1, …, v_n графа G, что для него выполняются следующие условия:

1. $v_1 = u_1, v_2 = u_2$;

2. Для $3 \le k \le n$ пусть подграф G_k будет плоский подграф графа G, порожденный вершинами v_1, …, v_k и пусть C_k будет внешней гранью G_k. Вершина v_k находится на грани C_k. Также, для $k < n$ вершина v_k имеет хотя бы одного соседа в $G - G_k$;

3. Для каждого $3 \le k \le n - 1$ подграф G_k есть двусвязный и *внутренне максимальный* (все внутренние грани G_k являются треугольниками);

4. $v_n = u_0$.

1.5.1 Метод сдвига

Алгоритм Фрейззе – Паха – Поллака представляет собой построение плоского изображения с прямыми ребрами n-вершинного максимального плоского графа на сетке размера $(2n - 4) \times (n - 2)$. Алгоритм сводится к следующему:

1. Вершины помещаются на сетку по одной в соответствии с каноническим порядком входного графа.

2. На каждом шаге контур изображения текущего графа сохраняет некоторые инварианты, включая ограничения на наклон ребер.

3. Когда вершина помещается на сетку, некоторые из ранее размещенных вершин сдвигаются налево, а некоторые – направо, чтобы сохранить инварианты и планарность текущего изображения.

1.6 Генерация случайного планарного графов

Алгоритм для генерации случайного связного плоского графа с n вершинами:

1. Зададим координаты границ двумерной прямоугольной области. Затем сгенерируем n случайных равномерно распределенных точек. После этого отсортируем список точек по одной из координат;

2. Если возможно, соедините как можно больше n точек непересекающимися прямыми линиями. Для это воспользуемся алгоритмом триангуляции Делоне. После применения триангуляции область будет триангулирована;

3. Случайное удаление ребер. На этом шаге следует попытаться последовательно удалить некоторое заранее определенное количество случайных ребер. Но перед удалением ребра, необходимо проверить:

3.1. является ли ребро мостом;

3.2. является ли хотя бы одна вершина ребра точкой сочленения;

3.3. какова степень вершин ребра.

2 Практическая часть

2.1 Программа для работы с планарными графами

Программа для работы с планарными графами была реализована в среде Microsoft Visual Studio на языке C++ CLI. Данная программа позволяет определить, является ли граф планарным с помощью Гама-алгоритма, визуализировать любой граф по окружности или с помощью использования физических аналогий, а также сгенерировать случайный планарный граф. Если граф является планарным и связным, то можно визуализировать его плоское изображение с помощью метода сдвига.

2.2 Примеры работы программы

Пример 1. Проверка планарного графа на планарность и его визуализация. Введем его с помощью матрицы смежности. Результат видим на рисунке 2. Граф по умолчанию отрисовывается по окружности.

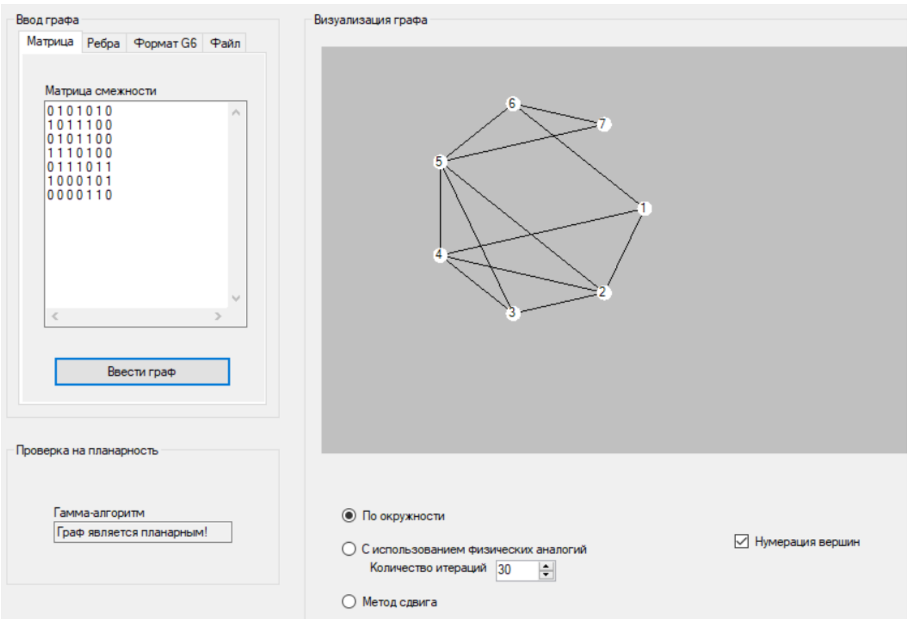


Рисунок 2 – Результат работы программы

Пример 2. Проверка на планарность графа, который не является планарным, и его визуализация. Теперь введем граф через ребра. Отрисуем граф с использованием физических аналогий, количество итераций равно 7. На рисунке 3 можем увидеть результат.

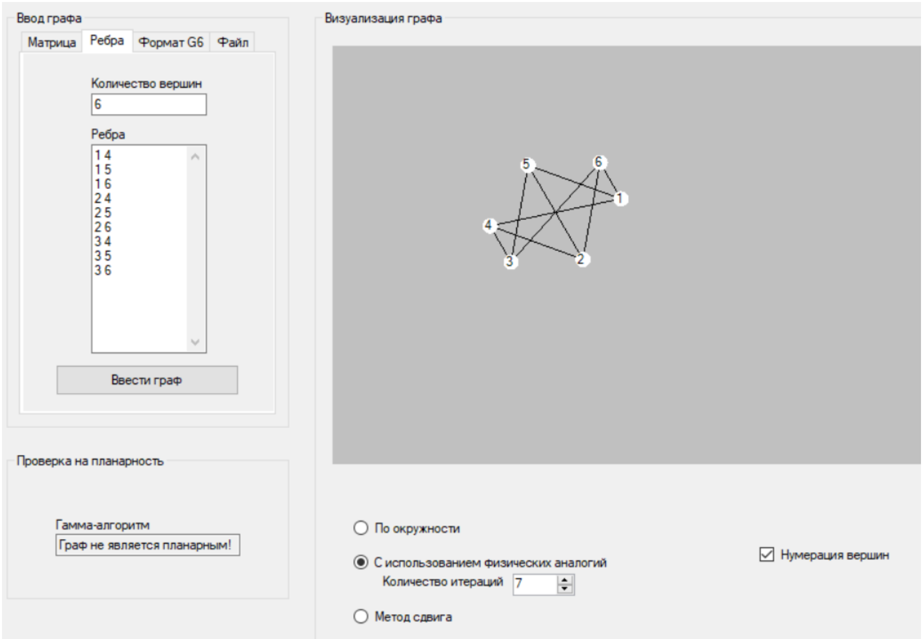


Рисунок 3 – Результат работы программы

2.3 Работа программы вместе с nauty

С помощью nauty будем генерировать графы с заданным числом вершин в формате graph6.

Пример 1. Проверка планарного графа на планарность и его визуализация. Теперь введем граф в формате *graph6*. Отрисуем граф методом сдвига. На рисунке 4 можем увидеть результат.

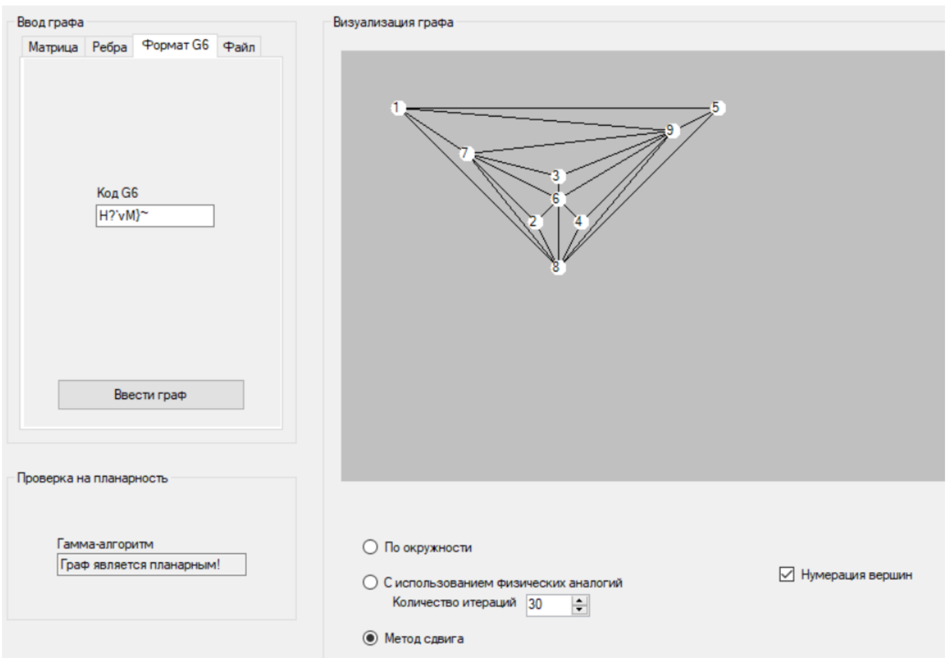


Рисунок 4 – Результат работы программы

Пример 2. Проверка графов на планарность и их визуализация. Теперь введем графы с помощью файла. Будем проверять графы с 9 вершинами. На рисунке 5 можем увидеть результат.

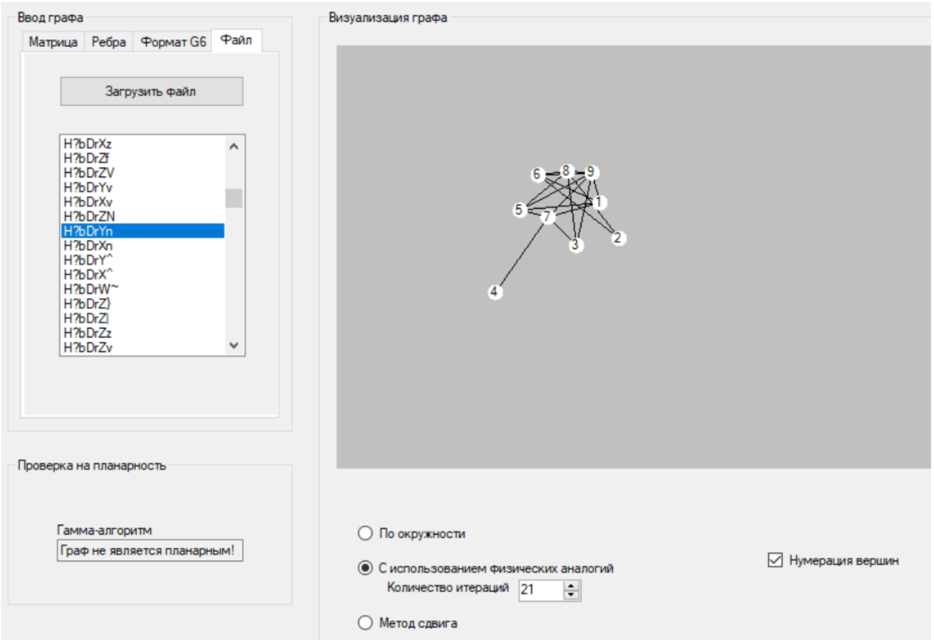


Рисунок 5 – Результат работы программы

2.4 Генерация случайного планарного графа

Пример 1. Сгенерируем планарный граф с 12 вершинами, случайным образом удалим 5 вершин. На рисунке 6 показана генерация.

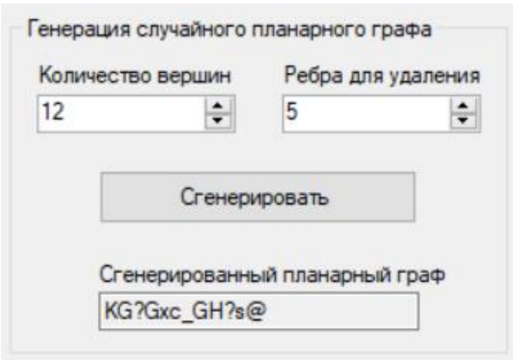


Рисунок 6 – Генерация планарного графа

Теперь на рисунке 7 проверим данный граф на планарность и отрисуем его. Данный граф является планарным.

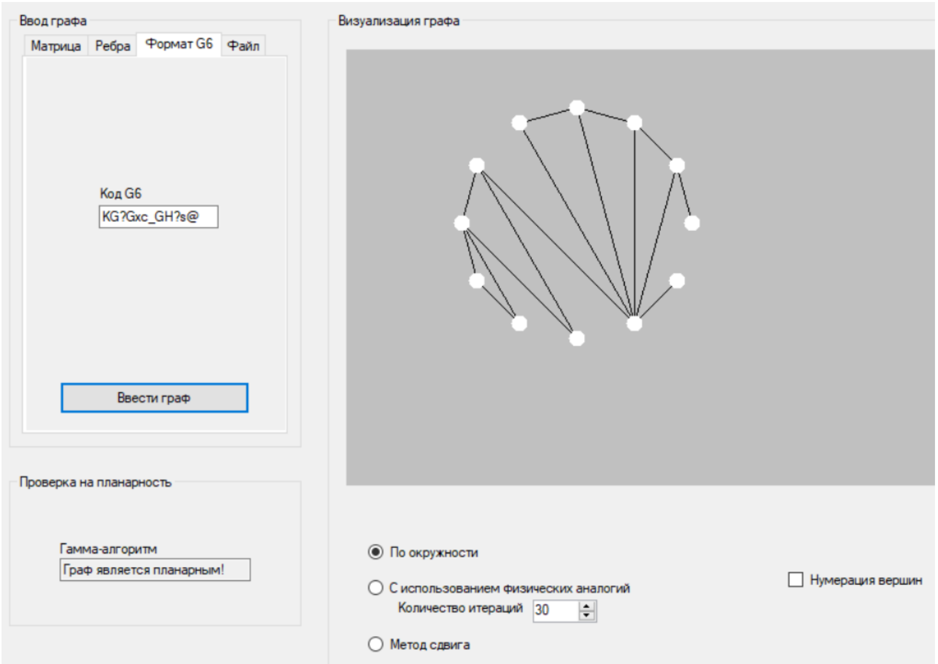


Рисунок 7 – Планарный граф

2.5 Исследование планарных графов

В результате исследования были получены результаты, которые были проверены с помощью The on-line encyclopedia of integer sequences (последовательности A005470, A000055) и Мира графов.

Теперь проанализируем все результаты. По таблице 1 можем заметить, что с увеличением числа вершин, количество планарных графов уменьшается.

Таблица 1 – Планарность графов в процентном соотношении

Количество вершин	Планарные графы	Непланарные графы	Подходящие под Гамма-алгоритм	Деревья	Несвязные	С мостами
6	91.02564%	8.97436%	30.76923%	3.84615%	27.56410%	50.00000%
7	78.73563%	21.26437%	31.32184%	1.05364%	16.85824%	40.99617%
8	56.42313%	43.57687%	25.43334%	0.18630%	8.03499%	27.66888%
9	29.07255%	70.92745%	14.18840%	0.01711%	2.90096%	13.56292%
10	9,50354%	90,49646%	4,79695%	0,00088%	0,73394%	4,34734%
11	1.83327%	98.16673%	0.92913%	0.00002%	0.12087%	0.84294%

ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен необходимый математический аппарат для работы с планарными графами. Мы рассмотрели алгоритм для проверки планарности графа – Гамма-алгоритм. Был рассмотрен алгоритм с использованием физических аналогий для визуализации графов. Также для визуализации планарных рассмотрели метод сдвига. Был рассмотрен алгоритм генерации случайных планарных графов.

Был разработан программный комплекс для работы с планарными графами на языке C++ CLI в среде Microsoft Visual Studio 2017. В данной программе были реализованы Гамма-алгоритм для проверки планарности, алгоритм для визуализации графов с использованием физических аналогий, визуализация планарных графов методом сдвига, генерация случайных планарных графов. Были проанализированы все графы с числом вершин до 11 и среди них найдены все планарные. Также было проверено какие из них подходят под непосредственное применение Гамма-алгоритма. Подтвердились известные результаты, что с увеличением числа вершин количество планарных графов уменьшается.