

Графы. BFS. Планарность

Гусев Илья, Булгаков Илья

Московский физико-технический институт

Москва, 2018

Содержание

- Графы. BFS. Приложения
- Планарные Графы
- Задача о плоской укладке.
- Гамма-алгоритм.

BFS. Поиск минимального цикла

Как найти минимальный цикл в ориентированном невзвешенном графе?

- Запустим поиск в ширину из каждой вершины;
- как только в процессе обхода мы пытаемся пойти из текущей вершины по какому-то ребру в уже посещённую вершину, то это означает, что мы нашли кратчайший цикл, и останавливаем обход в ширину;
- среди всех таких найденных циклов (по одному от каждого запуска обхода) выбираем кратчайший.

BFS. Проверка на двудольность

Как определить, является ли граф двудольным?

- Произведём серию поисков в ширину. Т.е. будем запускать поиск в ширину из каждой непосещённой вершины.
- Ту вершину, из которой мы начинаем идти, мы помещаем в первую долю. В процессе поиска в ширину, если мы идём в какую-то новую вершину, то мы помещаем её в долю, отличную от доли текущей вершины. Если же мы пытаемся пройти по ребру в вершину, которая уже посещена, то мы проверяем, чтобы эта вершина и текущая вершина находились в разных долях. В противном случае граф двудольным не является.
- По окончании работы алгоритма мы либо обнаружим, что граф не двудолен, либо найдём разбиение вершин графа на две доли.

BFS. Поиск количества минимальных путей между вершинами

Как найти количество различных кратчайших путей между заданными вершинами `src` и `dst`?

- Запускаем модифицированный BFS. Для каждой посещенной вершины храним длину минимального пути в нее и число уже найденных путей в нее.
- При каждом заходе в посещенную вершину обновляем ее "показатели". Если найден более короткий путь, то сбрасываем счетчики. Если путь длиннее - игнорируем его. Если путь такой же - увеличиваем счетчик числа путей.

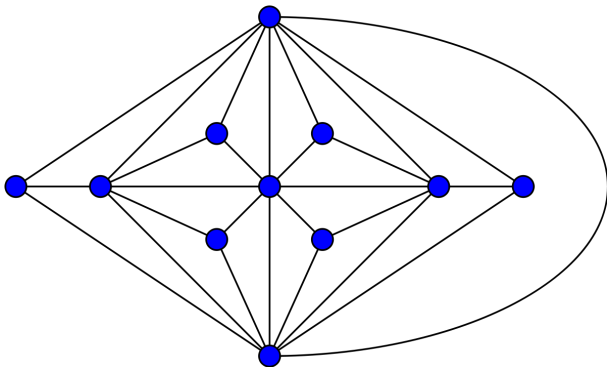
Планарный граф

Плоский граф — это граф, нарисованный таким образом, что его ребра не пересекаются.

Граф допускает **плоскую укладку**, если его можно нарисовать как плоский. Плоские графы называют **планарными**.

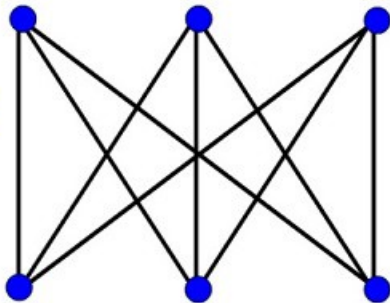
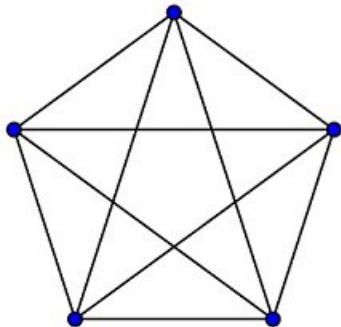
Пример планарного графа

Граф является планарным, так как его можно нарисовать так, чтобы его ребра не пересекались



Пример непланарного графа

Здесь показаны два непланарных графа: полный пятивершинник и полный двудольный граф. Для них есть специальные обозначения: K_5 и $K_{3,3}$ соответственно.



Задача о плоской укладке.

Задача: Определить, является ли граф планарным, и, если да, то произвести его плоскую укладку.

- Теория. Есть теорема Понтрягина-Куратовского, которая утверждает: граф планарен тогда и только тогда, когда он не содержит подграфов, гомеоморфных K_5 и $K_{3,3}$. Представляет теоретический интерес.
- Практика. Есть алгоритмы, которые позволяют построить плоскую укладку, если она существует. Рассмотрим **гамма-алгоритм**.

Гамма-алгоритм

Гамма-алгоритм: построение плоской укладки.

Входные данные:

- Граф связный. Если граф несвязный, но рассматриваем каждую компоненту связности. (Повторение: как найти компоненту связности?)
- Граф имеет хотя бы один цикл. Если граф - дерево, то он планарен.
- Граф не имеет мостиков, т. е. ребер, после удаления которых граф распадается на две компоненты связности. Если мосты есть, они убираются и компоненты укладываются по очереди: каждая последующая укладывается на правильной грани.

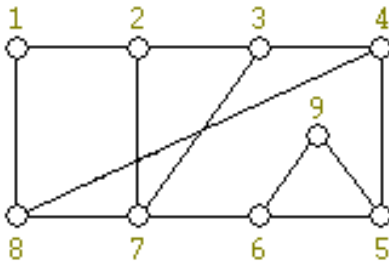
Гамма-алгоритм

Гамма-алгоритм: формальное описание

- **Инициализация:** Выберем любой простой цикл C исходного графа G ; изобразим его на плоскости в виде грани, которую примем за уже уложенную часть G^{plane} ; сформируем сегменты S_i ; если множество сегментов пусто, граф уложен
- **Шаг алгоритма.** Пока множество сегментов непусто:
 - Для каждого сегмента S найти множество $\Gamma(S)$. Если существует сегмент S , для которого $|\Gamma(S)| = 0$, то граф не планарный, конец.
 - Выбираем один из сегментов с минимальным числом вмещающих его граней.
 - Выбираем одну из подходящих граней для выбранного сегмента.
 - В данном сегменте выбираем цепь между двумя контактными вершинами и укладываем ее в выбранной грани. Учтем изменения в структуре сегментов. Переход к новому шагу
- **Завершение.** Построена плоская укладка G^{plane} исходного графа G .

Гамма-алгоритм

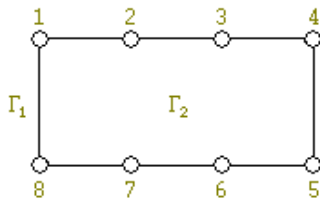
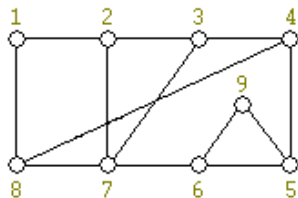
Дан связный граф с циклом и без мостов



Гамма-алгоритм

Инициализация. Находим любой простой цикл.

Получаем две грани: Γ_1 — внешнюю и Γ_2 — внутреннюю. Обозначим выбранный цикл как G^{plane} .

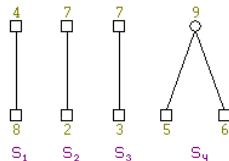
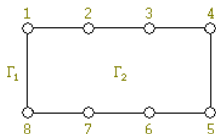
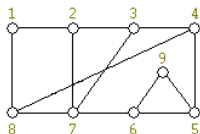


Гамма-алгоритм

Шаг алгоритма. Строим множество сегментов. Каждый сегмент S относительно G^{plane} представляет собой одно из двух:

- ребро, оба конца которого принадлежат G^{plane} , но само не принадлежит G^{plane} ;
- связную компоненту графа $G - G^{plane}$, дополненную всеми ребрами графа G , т.ч. один из концов принадлежит $G - G^{plane}$, а второй G^{plane}

Те вершины, которые одновременно принадлежат G^{plane} и какому-то сегменту, назовем контактными. Сегменты и вершины изображены на слайде справа. Контактные вершины обведены в квадрат.



Гамма-алгоритм

Утверждение.

В каждом сегменте не менее 2 контактных вершин

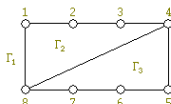
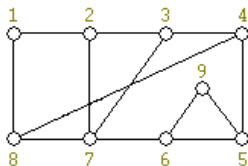
Если бы 0, то несвязный граф

Если бы 1, то мост

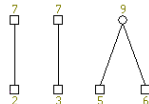
Значит, в любом сегменте есть цепь между любой парой контактных вершин

Гамма-алгоритм

Пока для любого i : $S_i \in \Gamma_1, \Gamma_2, |\Gamma(S_i)| = 2$. Поэтому возьмем первый по номеру сегмент S_i и в нем первую попавшуюся цепь $\{4, 8\}$; вставим эту цепь в грань Γ_2 . Получим увеличенную часть G^{plane} и уменьшенную систему сегментов



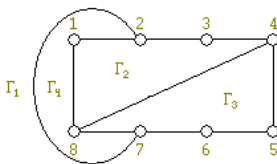
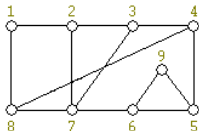
a)



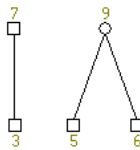
b)

Гамма-алгоритм

Определим, какие грани вмещают новые сегменты. Теперь сегменты S_1 и S_2 вмещаются только в одну грань Γ_1 , в то время, как сегмент S_3 вмещается в две грани Γ_1 и Γ_3 . Поэтому берем S_1 . Возьмем в нем цепь между контактными вершинами, например, $\{2, 7\}$, и уложим ее в Γ_1 . Получим увеличенную часть G^{plane} и уменьшенную систему сегментов



a)

 S_1

b)

 S_2

Гамма-алгоритм

Продолжая таким образом, в итоге получим плоскую укладку графа G

