

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

**(ДВФУ)**

|  |
| --- |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**  **Департамент математического и компьютерного моделирования** |

**ДОКЛАД**

**о практическом задание по дисциплине АИСД**

«Алгоритм Голдьберга-Тарьяна»

направление подготовки 09.03.03 «Прикладная информатика»

профиль «Прикладная информатика в компьютерном дизайне»

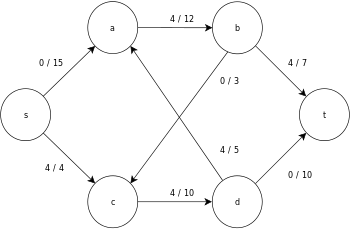
|  |  |  |
| --- | --- | --- |
|  |  | Выполнил студент  гр. Б9121-09.03.03пикд  Герасимов Евгений Александрович \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Доклад защищен:  С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | *(подпись)*  Руководитель практики  Доцент ИМКТ А.С Кленин  *(должность, уч. звание)*  \_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)*  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2022г. |
| Рег. № \_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |  |  |

г. Владивосток

2022

**Введение**

Для простоты понимания смоделируем ситуацию. Есть ациклический ориентированный граф. Ацикличность графа обусловлена тем, что такой граф может иметь ребра, выходящие из одной вершины и приходящие разными путями в конечный узел (сток). У графа есть две выделенные вершины: исток (S) и сток (T). Модель задачи такова: из истока в сток продвигается некоторое количество определенных элементов, проходя по ребрам графа. Для удобства будем называть эти условные элементы “жидкостью”. Жидкостью может быть что угодно: вода в канализационной системе или пропускная способность компьютерной сети. Для каждого ребра известно число: какое максимальное количество жидкости в секунду может по нему протечь (проходимость ребра не может быть меньше нуля). Ребро не может превысить свою максимальную пропускную способность. Жидкость течет в направлении стрелочек-указателей направления в случае, если граф ориентированный.



Имеется некий потоковый алгоритм, перед которым поставлена цель максимизировать проходимость графа и провести из истока в сток как можно больше жидкости. Убедиться в верности потока можно взяв вершину и убедившись в том, что попавшее в нее количество жидкости равняется вышедшему, это дает понять, что в вершине не задерживается и не появляется новая жидкость (данные, вода и другие ресурсы). Понять сколько жидкости передвигается из истока в сток (фактически, на выходе) можно, суммировав сколько жидкости прошло до стока по ведущим к нему ребрам.

Основные понятия. Введем несколько необходимых для понимания определений. Обозначим F (flow). Пусть F = (суммарное количество жидкости, вытекающее из S) = (суммарное количество жидкости, втекающее в T), где – поток, и – ребра, входящие в сток и ребра, выходящие из истока. Обозначим также поток по ребру как а максимальную пропускную способность как , где – ребро графа (дополняя, обратным данному ребру будет ребро ).

Введем понятие разрез. Смоделируем граф, на каждом ребре которого вновь указано . Множество ребер, удаление которых приведет к тому, что путь из истока в сток пропадет, назовем C. Величина разреза равна суммарной стоимости удаленных ребер. Логически следует что максимальный поток равен минимальному C.

Возьмем какой-то путь, вдоль которого увеличим поток (добавим к существующему потоку некоторое допустимое количество). Чтобы пути вследствие увеличения потока оставались правильными введем понятие обратное ребро. Обратное ребро имеет поток равный отрицательному потоку исходного ребра. Теперь сформулируем задачу потокового алгоритма в данной модели согласно следующим законам: (для всех вершин, за исключением истока и стока); – ?

Еще одной важной частью работы эффективных потоковых алгоритмов является так называемая остаточная сеть. Остаточная сеть обозначается как: . Остаточная сеть дает понять на сколько можно увеличить поток вдоль ребра.

На данном этапе можно приступить к рассмотрению алгоритма Диница – основы алгоритма Гольдберга-Тарьяна. Данный алгоритм находит путь используя DFS, на этом пути находится ребро с минимальной пропускной способностью и удаляется, затем поток вдоль пути увеличивается на пропускную способность ребра. Будет не лишним еще раз объяснить, почему же алгоритм действует именно так. Все очень просто: мы не можем увеличить поток больше, чем может пропустить самое “слабое” ребро. Может возникнуть ситуация, когда путь не справится с увеличенным потоком, в таком случае поток нужно уменьшить. В конечном счете алгоритм придет к тому, что на каждом возможном пути уже нельзя будет беспрепятственно увеличить поток. Поток, который приводит к тому, что любой путь из истока в сток уже имеет ребро, выдающее свою максимальную пропускную способность, называется блокирующим. Блокирующие потоки являются ключевой частью алгоритма. Найденные блокирующие потоки прибавляются к текущему потоку. В конечном счете алгоритм узнает максимальную пропускную способность графа.

Несмотря на то, что данный алгоритм прост и эффективен, он медленнее чем другие алгоритмы (вариации). Основная проблема заключается в том, что поиск блокирующих потоков может крайне сильно отличаться асимптотической сложностью. Вариация, где алгоритм ищет пути с помощью DFS имеет сложность , где m количество путей, и всего таких путей будет не больше , так как каждая итерация-путь дает максимум нагрузки хотя бы одно ребро. Итоговая асимптотическая сложность равна .

В другой вариации, которая принята как стандарт, лишние ребра удаляются. Для этой цели строится так называемая слоистая сеть. Определяются длины кратчайших путей из истока до всех остальных вершин, в слоистую сеть включаются все ребра исходной сети, которые ведут на более глубокий уровень, проверяется это инкрементацией индекса глубины. В слоистой сети, конечном виде, каждый путь является кратчайшим. В такой сети алгоритм тратит n на путь. Ребер m значит и путей m. Всего может быть n фаз этого алгоритма, следовательно асимптотическая сложность составляет .

**Реализация алгоритма**

Однако, можно пойти дальше, и применив алгоритм Гольдберга-Тарьяна можно достичь асимптотической сложности . Идея поиска пути с самого начала с каждой итерацией может быть и проста в реализации, но не дает возможности использовать уже пройденные части предыдущих путей для более быстрого их прохождения. Если продолжать движение по путям, которые уже были пройдены, алгоритм вероятно приведет к разрезу. После того как лишние ребра графа были отрезаны вместо графа остается несколько отдельных деревьев, которые называются лесом. При этом фактически, несмотря на “вырезанные” ребра, деревья до сих пор с друг другом соединены, и на самом деле лишь демонстрируют уже пройденные отрывки пути. Вершина S при этом осталась в каком-то из деревьев. Далее соединяются корни дерева с ребрами других деревьев, начиная с самого “высокого”. Затем находится путь от истока в сток, на нем находится минимальное ребро, пускается добавочный поток равный пропускной способности минимального ребра. Минимальное ребро вырезается.

Данные действия позволяют отсортировать лес по высоте, вследствие чего находится максимально эффективный путь.

while BFS:

while True:

r = root(s)

if r = t:

delta = min (st) //минимум на пути ст

f += delta на пути st //увеличить С штрих (поток)

(uv) – min ребро на пути st

удалить ребро (uv)

else:

if есть ребро r -> v (rv)

добавить ребро (rv) в лес //процедура ищет корень

else удалить r вместе с ребрами

if r == t break

**Источники**

**[1] Математическая лаборатория имени П. Л. Чебышева СПбГУ. Первая лекция на тему “Тестирующие и потоковые алгоритмы для слов, деревьев и графов”. Лектор: Michel de Rougemont.**

**[2] Университет ИТМО. Пятая лекция четвертого семестра на тему: “Потоки. Алгоритмы Гольдберга-Тарьяна и Хопкрофта-Карпа”. Лектор: Павел Маврин. 2020 г.**

**[3] Университет ИТМО. Пятнадцатая лекция четвертого семестра на тему: “Потоковые алгоритмы”. Лектор: Павел Маврин. 2021 г.**

**[4] Президентский физико-математический лицей номер 239. Двенадцатая из серии лекций по теме: “Дополнительные главы алгоритмов”. Лектор: Андрей Станкевич.**

**[5] Курс “Basic Programming”. Второй модуль, посвященный алгоритмам и структурам данных.**

**[6] Викиконспект университета ИТМО. Конспект на тему: “Алгоритм Голдберга-Тарьяна”.**

**[7] Викиконспект университета ИТМО. Конспект на тему: “Схема алгоритма Диница”.**

**[8] Книга “Алгоритмы. Построение и анализ” третье издание. Авторы: Штайн Клиффорд, Рональд Л. Ривест, Томас Х. Кормен. Издательство “Вильямс” 2007 г. Раздел: “Алгоритмы для работы с графами”. Глава: “Задача о максимальном потоке”.**