

# Лабораторная работа №9 по курсу дискретного анализа: Поиск максимального потока алгоритмом Форда-Фалкерсона

Выполнил студент МАИ группы М8О-201Б *Ефимов Александр*.

## Постановка задачи

Разработать программу на языке C или C++, реализующую указанный алгоритм согласно заданию:

Задан взвешенный ориентированный граф, состоящий из  $n$  вершин и  $m$  ребер. Вершины пронумерованы целыми числами от 1 до  $n$ . Необходимо найти величину максимального потока в графе при помощи алгоритма Форда-Фалкерсона.

Для достижения приемлемой производительности в алгоритме рекомендуется использовать поиск в ширину, а не в глубину.

Источком является вершина с номером 1, стоком – вершина с номером  $n$ .

Вес ребра равен его пропускной способности.

Граф не содержит петель и кратных ребер.

## Метод решения

При имплементации алгоритма Форда-Фалкерсона можно использовать его конкретизацию – алгоритм Эдмондса – Карпа, который при поиске использует поиск в ширину для нахождения кратчайшего пути. На бумаге такой алгоритм работает за  $O(|V| * |E|^2)$ , где  $|V|$  – количество вершин, а  $|E|$  – количество граней.

Для каждого найденного кратчайшего пути от начальной вершины до конечной ищется его минимальная пропускная способность: наименьший вес среди всех граней этого пути. Этот наименьший вес прибавляется к отдельной переменной, которая будет содержать максимальный поток графа, после чего он вычитается из всех весов в направлении от начала пути до его конца и прибавляется ко всем весам от конца пути до его начала.

Алгоритм продолжается до тех пор, пока существуют пути от начальной вершины до конечной, после чего возвращается найденная сумма всех весов.

## Описание программы

1. *StaticQueue.h* Шаблонная реализация очереди на массиве типа `std::array`
2. *FlowNetwork.h* Класс, содержащий в себе объявления самого графа в виде вектора векторов и двух методов:
  - *FindMaxFlow* – Находит максимальный поток графа

- *FindPath* – Находит путь между двумя вершинами графа.

### 3. *FlowNetwork.cpp* Реализация методов класса:

- *FindMaxFlow* – ищет путь между первым и последним вершинами графа. Если он не найден, то возвращает текущий максимальный поток (в самом начале равен нулю и, если путь не найден самого начала, то максимальный поток равен нулю). Если он найден, то ищет наименьший вес среди всех граней на этом пути, прибавляет этот вес к максимальному потоку, вычитает из всех граней в направлении конца этот вес и прибавляет к всем граням в направлении начала.
- *FindPath* – Используя очередь, проходит в ширину по всем вершинам графа, пока либо очередь не пуста, либо не найдена необходимая вершина, при этом указывает в два дополнительных массива посещенные вершины и их родителей.

4. *main.cpp* Содержит все функцию, принимающую данные из ввода и строит на основе них график, после чего ищет его максимальный поток.

## Дневник отладки

Номер	Ошибка	Обнаруженная причина	Исправление
1	Неправильный ответ	Переполнение переменной, содержащей максимальный поток	Увеличить размер переменной

## Тест производительности

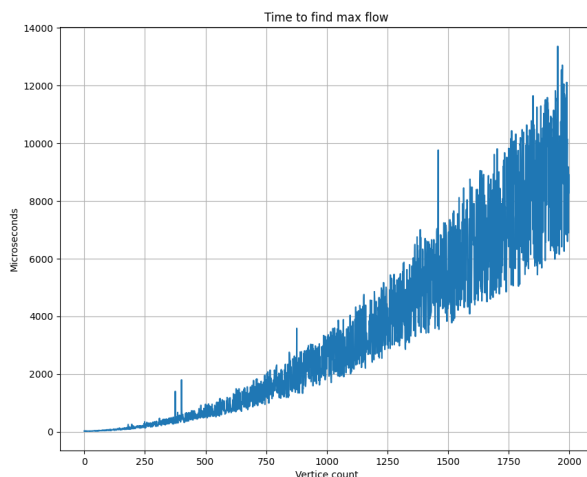


Рис. 1: Дерево, построенное на строке из повторяющихся символов в порядке алфавита

На рисунке показано время в микросекундах, которое ушло на нахождение максимального потока. По абсциссе показано количество вершин. Максимальное количество граней может быть в четыре раза больше вершин.

## Выводы

По теореме Форда-Фалкерсона, максимально возможный поток между двумя вершинами равен сумме всех весов границей таких, что, при исключении их из графика, между источником и стоком не останется путей, соединяющих их.

На основе этой теоремы Форда-Фалкерсона разработал идею алгоритма со сложностью  $O(|E| \cdot f^*)$  ( $f^*$  - максимальный поток на графе), позволяющего определить максимальную пропускную способность между двумя вершинами графа. Это полезно там, где нужно по каким-либо причинам определить максимально возможный поток ресурса (пакетов в сети, транспорта). Более специализированным примером будет нахождение максимального паросочитания на графе, но для этого нужно знать разделение этого графа на двудольный.

Но сам его алгоритм всего лишь идея, так как он не уточняет каким методом нужно находить путь между источником и стоком. Для его реализации нужно выбрать метод нахождения пути, например с помощью поиска в глубину или поиска в ширину. На поиске в ширину уже основывается алгоритм Эдмондса — Карпа, находящего максимальный поток за  $O(|V|^2 \cdot |E|)$