МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Построение и анализ алгоритмов»

Тема: Потоки в сети

Студент гр. 8303 _____ Пушпышев А.И.

Преподаватель _____ Фирсов А.М.

Санкт-Петербург

2020

Цель работы.

Реализовать алгоритм Форда-Фалкерсона, найти максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро.

Вар. 2. Поиск в ширину. Обработка совокупности вершин текущего фронта как единого целого, дуги выбираются в порядке уменьшения остаточных пропускных способностей. При равенстве остаточных пропускных способностей выбирается та дуга, из начала которой выходит меньше дуг, при этом учитываются только дуги с положительными остаточными пропускными способностями, не ведущие в вершины текущего или прошлых фронтов.

Задание.

Найти максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро, используя алгоритм Форда-Фалкерсона.

Сеть (ориентированный взвешенный граф) представляется в виде триплета из имён вершин и целого неотрицательного числа - пропускной способности (веса).

Входные данные:

N – количество ориентированных рёбер графа

V_0 – исток

V_N – сток

V_i V_j W_{ij} – ребро графа

V_i V_j W_{ij} – ребро графа

…

Выходные данные:

P_{max} – величина максимального потока

V_i V_j W_{ij} – ребро графа с фактической величиной протекающего потока

V_i V_j W_{ij} – ребро графа с фактической величиной протекающего потока

…

В ответе выходные рёбра отсортируйте в лексикографическом порядке по первой вершине, потом по второй (в ответе должны присутствовать все указанные входные рёбра, даже если поток в них равен 0).

Пример входных данных

7
a
f
a b 7
a c 6
b d 6
c f 9
d e 3
d f 4
e c 2

Пример выходных данных

12
a b 6
a c 6
b d 6
c f 8
d e 2
d f 4
e c 2

Описание алгоритма.

В начале работы алгоритма создается граф. С помощью поиска в ширину производится поиск пути от истока в сток. Если такой путь существует, то мы в этом пути ищем дугу с наименьшей пропускной способностью, затем обновляем граф: для всех дуг, которые попали в путь от истока в сток и для обратных им пересчитывается их пропускная способность. Затем производиться подсчет максимального потока, к этому мы шагу сразу переходим, если путь не был найден. Выводятся фактические потоки для дуг.

Сложность алгоритма по операциям: O (E * F), E – число ребер в графе, F – максимальный поток

Сложность алгоритма по памяти: O (N+E), N – количество вершин, E – количество ребер

Описание функций и структур данных.

```
class FordFalk {
	std::map< char, std::map< char, int > > graph;
}
```

Структура данных, используемая для хранения направленного графа. Представляет собой ассоциативный контейнер хранения вершин и соответствующего ей контейнера вершина-расстояние. Для каждой вершины хранится ассоциативный массив вершин, до которых можно добраться из текущей и вес пути до них.

1. `void maxFlow(char source, char estuary);`

   Метод вычисления максимального потока.

2. `int get_min_flow(std::map< char, std::pair< char, int > > way, char source, char stock);`

   Метод нахождения минимальной пропускной способности на пути.

3. `void update_net(std::map< char, std::map< char, int > > &network, std::map< char, std::pair< char, int > > way, char source, char stock);`

   Метод перестройки сети в соответствии с правилом: если дуга входит в путь, то к ней прибавляется обратная(относительно нуля)

величина минимальной пропускной способности в данном пути. Если дуга обратная той, которая находится в этом пути, то к ней прибавляется величина минимальной пропускной способности.

4. `std::map< char, std::pair< char, int > > BFS(std::map< char, std::map< char, int > > net, char source, char stock);`

   Поиск в ширину.

Тестирование и исследование.

Input	Output	Время
7 a f a b 7 a c 6 b d 6 c f 9 d e 3 d f 4 e c 2	12 a b 6 a c 6 b d 6 c f 8 d e 2 d f 4 e c 2	3 сек
10 a h a b 5 a c 4 a d 1 b g 1 c e 2 c f 3 d e 6 e h 4 f h 4 g h 8	6 a b 1 a c 4 a d 1 b g 1 c e 2 c f 2 d e 1 e h 3 f h 2 g h 1	2 сек
d a b 1000 a c 1000	2000 a b 1000 a c 1000	1 сек

b c 1 b d 1000 c d 1000	b c 0 b d 1000 c d 1000	

Пример изображения потока

d

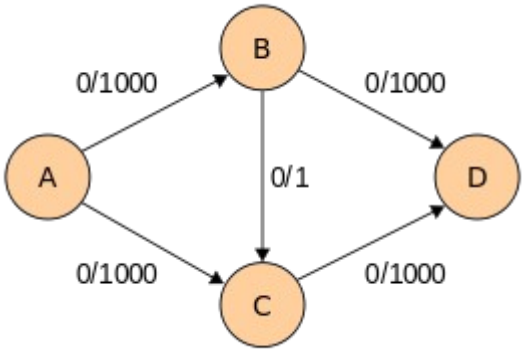
a b 1000

a c 1000

b c 1

b d 1000

c d 1000



2000

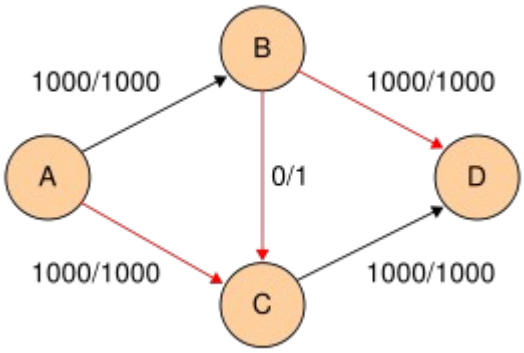
a b 1000

a c 1000

b c 0

b d 1000

c d 1000



Выводы.

В ходе выполнения лабораторной работы был реализован алгоритм Форда-Фалкерсона(точнее лгоритм Эдмондса — Карпа, так как используется поиск в ширину), который находит максимальный поток в сети, а также фактическую величину потока, протекающего через каждое ребро.