

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

**Алгоритмы и структуры данных
КДЗ 1**

Выполнил Днепров Данила,
студент гр. БПИ172.

Преподаватель:
Ахметсафина Римма Закиевна

Москва 2019



Постановка задачи

Разработать с использованием языка C++ программу, рассчитывающую максимальный поток в транспортной сети. Сделать это необходимо разными алгоритмами, такими как:

1. Базовая реализация метода Форда-Фалкерсона (Ford–Fulkerson algorithm),
2. Метод Форда-Фалкерсона в версии Эдмондса-Карпа (Edmonds–Karp algorithm),
3. Алгоритм Ефима Диница (Dinic's/Dinitz's algorithm).

Также необходимо провести экспериментальное исследование эффективности алгоритмов с целью оценки их быстродействия на разных входных данных. Для проведения эксперимента с алгоритмами нахождения максимального потока необходимо использовать набор из 40 txt-файлов с входными данными - таблицами смежности, где в названии содержится информация о графе:

1. В файлах с суффиксом 0.0 хранятся разреженные графы.
2. В файлах с суффиксом 0.5 хранятся графы средней плотности.
3. В файлах с суффиксом 1.0 хранятся плотные графы.
4. В файлах с суффиксом disco хранятся несвязные (disconnected) графы.

Таким образом, необходимо вычислить значения максимального потока в заданной транспортной сети, время, затраченное на вычисление значения максимального потока для каждого заданного графа тремя разными, вышеуказанными алгоритмами. Среднее время работы алгоритмов для каждой заданной транспортной сети. Также по итогам работы необходимо предоставить отчет содержащий постановку задачи, описание алгоритмов и задействованных структур данных, описание реализации, обобщенные результаты измерения эффективности алгоритмов, описание использованных инструментов (например, если использовались скрипты автоматизации), оценку соответствия результатов экспериментальной проверки теоретическим оценкам эффективности исследуемых алгоритмов.

Описание алгоритмов и использованных структур данных

Базовая реализация метода Форда-Фалкерсона

1. Обнуляем все потоки. Остаточная сеть изначально совпадает с исходной сетью.
2. В остаточной сети находим любой путь из источника в сток. Если такого пути нет, останавливаемся.
3. Пускаем через найденный путь максимально возможный поток:
 - а. На найденном пути в остаточной сети ищем ребро с минимальной пропускной способностью.



б. Для каждого ребра на найденном пути увеличиваем поток на найденное значение, а в противоположном ему — уменьшаем.

с. Модифицируем остаточную сеть. Для всех рёбер на найденном пути, а также для противоположных им рёбер, вычисляем новую пропускную способность. Если она стала ненулевой, добавляем ребро к остаточной сети, а если обнулилась, стираем его.

4. Возвращаемся на шаг 2.

Метод Форда-Фалкерсона в версии Эдмондса-Карпа (Edmonds–Karp algorithm)

В данной реализации метода Форда-Фалкерсона выбранный путь является кратчайшим, найденным с помощью поиска в ширину. Остальной алгоритм остается тем же.

Алгоритм Ефима Диница (Dinic's/Dinitz's algorithm).

1. Строится уровневый граф — каждой вершине определяется уровень в зависимости от числа ребер от истока.

2. Поиск блокирующего потока

а. Искать s - t пути по одному, пока такие пути находятся с помощью поиска в глубину.

б. Во время шага а удалять все ребра, по которым мы не можем дойти до вершины.

3. Найденный блокирующий поток прибавляется к текущему потоку.

4. Итерации суммируются (пока есть блокирующие потоки).



Результаты экспериментов Получившиеся ответы

File Name	Answer
input_10_0.0.txt	31
input_10_0.5.txt	45
input_10_1.0.txt	486
input_10_disco.txt	338
input_310_0.0.txt	63
input_310_0.5.txt	7817
input_310_1.0.txt	16157
input_310_disco.txt	7656
input_610_0.0.txt	87
input_610_0.5.txt	16592
input_610_1.0.txt	33270
input_610_disco.txt	14500
input_910_0.0.txt	150
input_910_0.5.txt	23492
input_910_1.0.txt	49212
input_910_disco.txt	24059
input_1210_0.0.txt	140
input_1210_0.5.txt	31993
input_1210_1.0.txt	65717
input_1210_disco.txt	30665
input_1510_0.0.txt	88
input_1510_0.5.txt	41844
input_1510_1.0.txt	80504
input_1510_disco.txt	38754
input_1810_0.0.txt	122
input_1810_0.5.txt	49487
input_1810_1.0.txt	99508
input_1810_disco.txt	48794
input_2110_0.0.txt	67
input_2110_0.5.txt	58419
input_2110_1.0.txt	111565
input_2110_disco.txt	57884
input_2410_0.0.txt	42
input_2410_0.5.txt	64995
input_2410_1.0.txt	129297
input_2410_disco.txt	62977
input_2710_0.0.txt	134
input_2710_0.5.txt	74659
input_2710_1.0.txt	145476
input_2710_disco.txt	72513



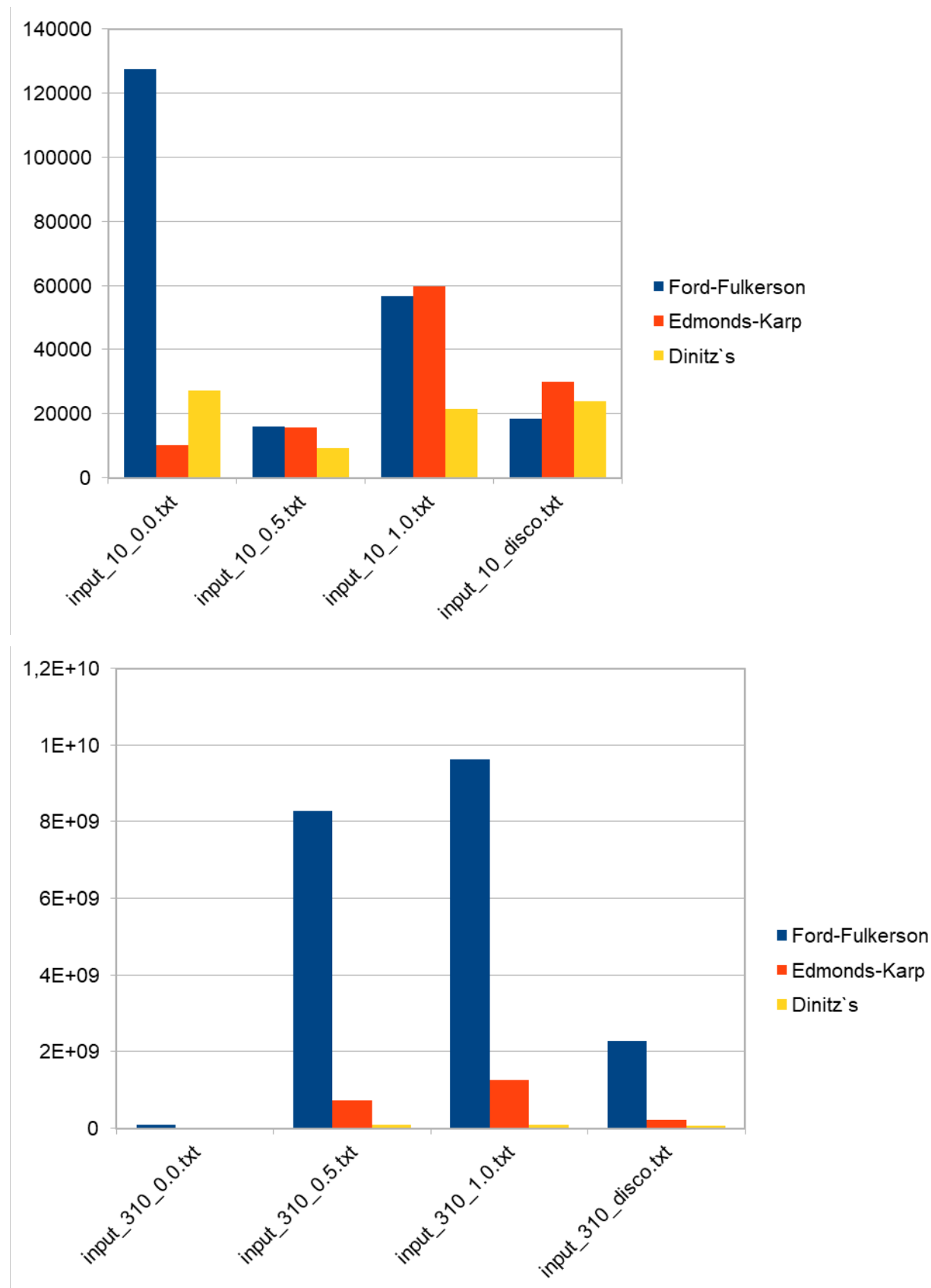
Результаты экспериментов

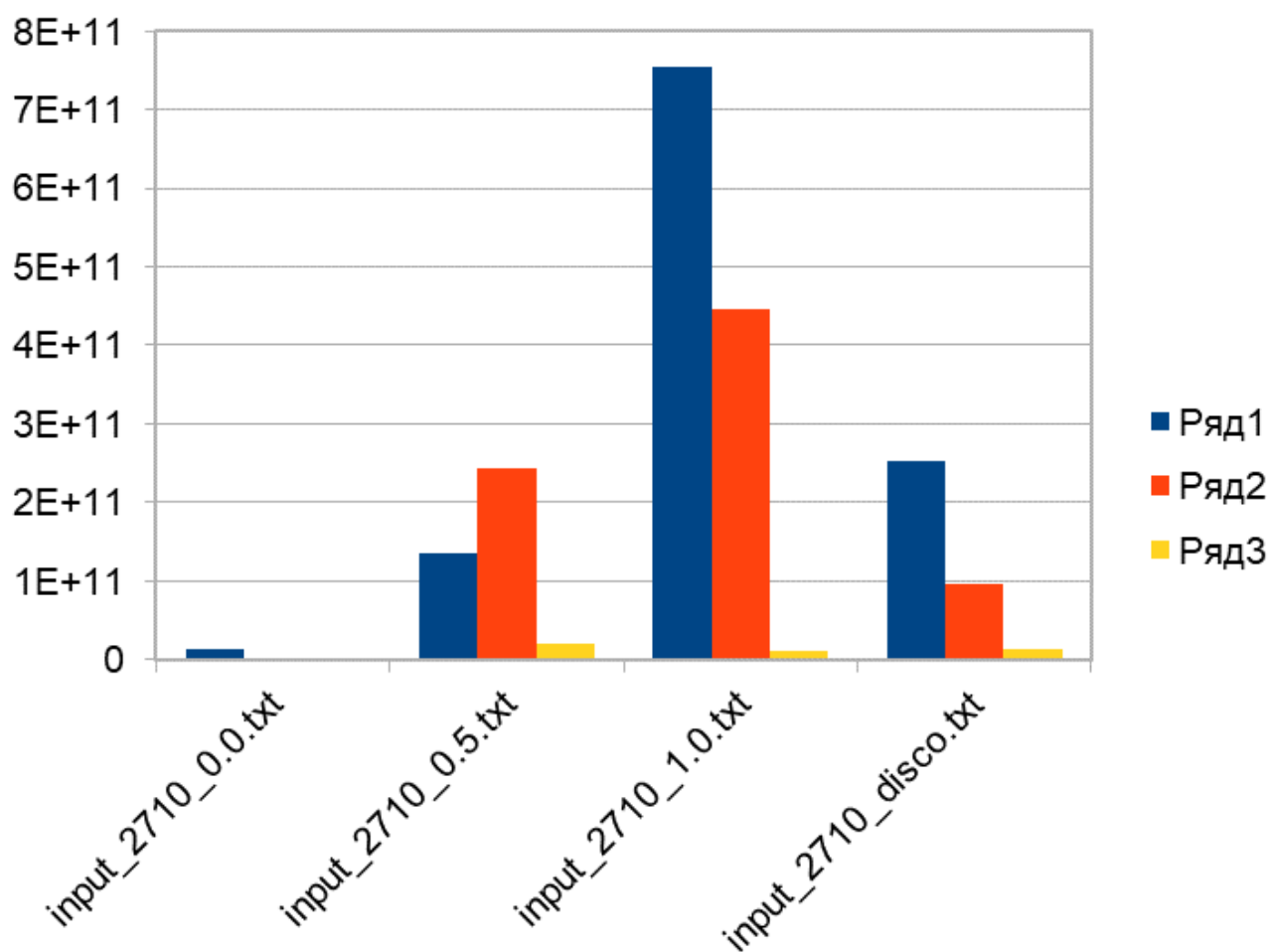
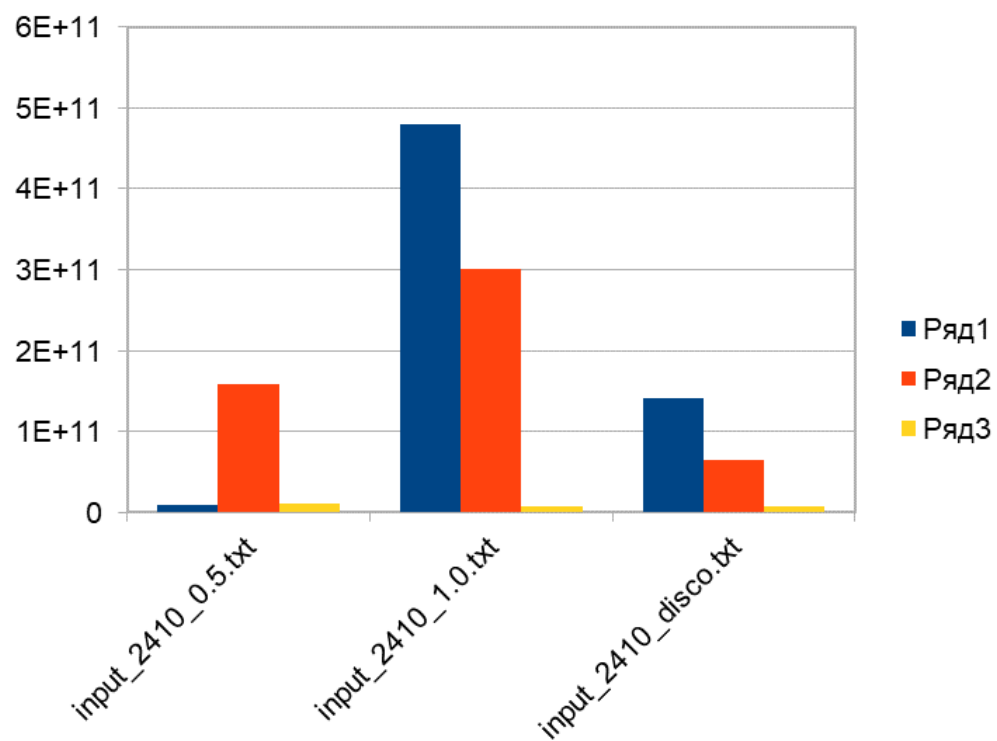
Среднее время работы алгоритмов (в наносекундах)

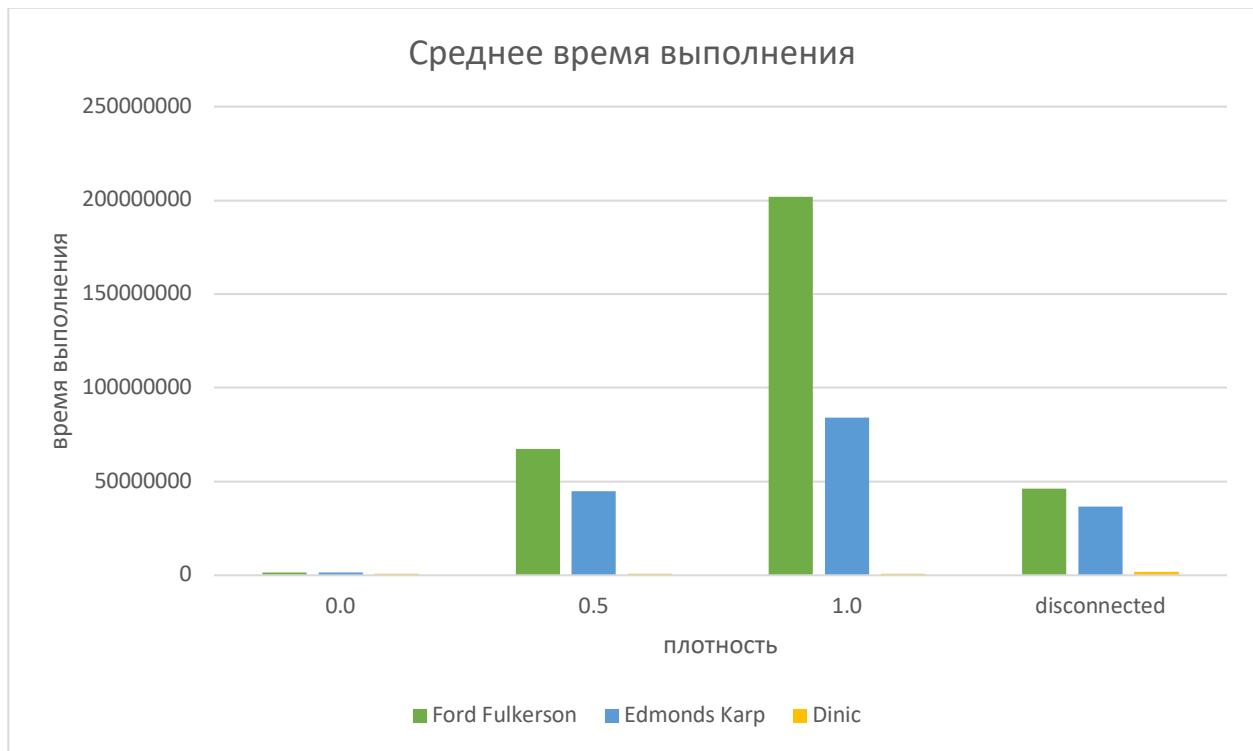
File Name	Ford-Fulkerson	Edmonds-Carp	Dinitz's algorithm
input_10_0.0.txt	18722	14075	27173
input_10_0.5.txt	13700	9331	9375
input_10_1.0.txt	63418	75038	21465
input_10_disco.txt	33405	32318	23830
input_310_0.0.txt	42104812	2064610	15343968
input_310_0.5.txt	1547875453	26915557	82330890
input_310_1.0.txt	2833859072	50206715	92756710
input_310_disco.txt	532703688	11529195	68459270
input_610_0.0.txt	167764472	15569946	56866514
input_610_0.5.txt	11548715750	213452500	543455720
input_610_1.0.txt	18188520550	370657736	623682453
input_610_disco.txt	1878725784	79794304	3096208574
input_910_0.0.txt	300146055	40870593	147476189
input_910_0.5.txt	16362926530	691890894	202057100
input_910_1.0.txt	34086839732	1221823842	1185641901
input_910_disco.txt	7283208440	279485594	1225030100
input_1210_0.0.txt	507295457	68812000	600911381
input_1210_0.5.txt	38541222015	1573745044	4427265960
input_1210_1.0.txt	78249778212	2845729806	1580262550
input_1210_disco.txt	16169446517	615474495	1783651630
input_1510_0.0.txt	479271968	75533965	355200804
input_1510_0.5.txt	78637494496	3133503735	7449399770
input_1510_1.0.txt	151576661613	5424765601	2770412010
input_1510_disco.txt	32212137099	1267092669	3265140990
input_1810_0.0.txt	1005124365	97280704	458433700
input_1810_0.5.txt	132334079205	5253581945	10336703760
input_1810_1.0.txt	273126038270	9513085596	5427784230
input_1810_disco.txt	64046544785	2292924808	5547541700
input_2110_0.0.txt	763926171	121841198	658186253
input_2110_0.5.txt	217300833015	8515623869	8797844105
input_2110_1.0.txt	400870477042	14930900241	5358950207
input_2110_disco.txt	101948578007	3555865999	6618573401
input_2410_0.0.txt	538532184	115016768	539763439
input_2410_0.5.txt	307553552570	12930653289	11182440100
input_2410_1.0.txt	606595518946	22188174746	8177129609
input_2410_disco.txt	143695604044	5147902625	7580890407
input_2710_0.0.txt	2409405918	241233392	1029913419
input_2710_0.5.txt	448197425334	17713631064	19119602605
input_2710_1.0.txt	861977940804	32043028875	11177753907
input_2710_disco.txt	207255812458	7456561880	12844586908



Компьютер на котором проводились испытания - MacBook Air (13-inch, Early 2015). Процессор - 1,6 GHz Intel Core i5. IDE – Clion.







Сравнительный анализ алгоритмов

Сравним теоретическую сложность через O асимптотику:

Сложность кодирования:

1. Базовая реализация метода Форда-Фалкерсона, сложность алгоритма: $O(|V||E|^2)$
2. Метод Форда-Фалкерсона в версии Эдмондса-Карпа, сложность алгоритма: $O(|V||E|^2)$
3. Алгоритм Ефима Диница сложность алгоритма: $O(|V|^2|E|)$

По результатам эксперимента легко заметить, что алгоритм Диница является универсальным алгоритмом, который может успешно применяться к практически любым графам.



Выводы

На больших и средних графах однозначно следует применять алгоритм Диница.

Источники

<http://e-maxx.ru/algo/dinic>

Седжвик Р. – Фундаментальные алгоритмы на C++. Части 1-4 (2001)

<https://www.youtube.com/watch?v=M6cm8Ueezil&t=192s>

<https://www.geeksforgeeks.org>

