

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Алгоритмы и структуры данных»

Выполнил: Кумар Денис

Студент гр. БПИ172

Преподаватель: Ахметсафина Римма Закиевна

## Оглавление

1.Постановка задачи.....	3
1.1.Постановка задачи о нахождении максимального потока .....	3
2.Используемые алгоритмы и структуры данных.....	3
2.1.Алгоритмы поиска максимального потока .....	3
2.2.Структуры данных, использованные для эксперимента .....	3
3.Описание плана эксперимента.....	3
3.1.Реализация эксперимента .....	3
4.Результаты эксперимента .....	3
4.1.Результаты вычислений .....	3
4.2.Результаты измерений времени .....	3
4.3.Графики времени работы разных алгоритмов на одном файле.....	3
4.4.Графики времени работы одного алгоритма на разных файлах .....	3
5.Сравнительный анализ алгоритмов.....	3
5.1.Анализ данных на основе результатов эксперимента .....	3
5.2.Асимптотические оценки времени работы алгоритмов.....	3
6.Заключение .....	3
7.Использованная литература .....	3

## 1. Постановка задачи

Задача контрольного домашнего задания заключалась в реализации трёх алгоритмов расчета максимального потока в транспортной сети, проведении экспериментального исследования их эффективности и анализе полученных результатов.

**Определение 1.** Задача о максимальном потоке заключается в нахождении такого потока по транспортной сети, что сумма потоков из истока, или, что то же самое, сумма потоков в сток максимальна.

**Определение 2.** В теории графов транспортная сеть — ориентированный граф, в котором каждое ребро имеет неотрицательную пропускную способность и поток. Выделяются две вершины: источник и сток такие, что любая другая вершина сети лежит на пути из источника в сток.

**Поток (flow)** — функция со следующими свойствами для любых вершин:

- **Ограничение пропускной способности (capacity constraints).** Поток не может превысить пропускную способность.  $f(u, v) \leq c(u, v)$
- **Антисимметричность (skew symmetry).** Поток из  $u$  в  $v$  должен быть противоположным потоку из  $v$  в  $u$ .  $f(u, v) = -f(v, u)$
- **Сохранение потока (flow conservation):**  $\sum_{w \in V} f(u, w) = 0$  для всех  $u \in V$ , кроме источника и стока.

**Величина потока (value of flow)** — сумма потоков из источника.

## 2. Используемые алгоритмы и структуры данных

### 2.1. Алгоритмы поиска максимального потока

В данной задаче рассматриваются 3 алгоритма поиска максимального потока в транспортной сети: Форда-Фалкерсона, Эдмонса Карпа и Диница. Идея алгоритма Форда-Фалкерсона заключается в следующем. Изначально величине потока присваивается значение 0: для всех. Затем величина потока итеративно увеличивается посредством поиска увеличивающего пути (путь от истока к стоку, вдоль которого можно послать больший поток). Процесс повторяется, пока можно найти увеличивающий путь. Суть алгоритма Эдмонса Карпа отличается от Форда-Фалкерсона только поиском увеличивающего пути (он находится поиском в ширину).

Если говорить о Динице, то алгоритм представляет собой несколько фаз. На каждой фазе сначала строится остаточная сеть, затем по отношению к ней строится слоистая сеть (обходом в ширину), а в ней ищется произвольный блокирующий поток. Найденный блокирующий поток прибавляется к текущему потоку, и на этом очередная итерация заканчивается. Этот

алгоритм схож с алгоритмом Эдмондса-Карпа, но основное отличие можно понимать так: на каждой итерации поток увеличивается не вдоль одного кратчайшего пути, а вдоль целого набора таких путей (ведь именно такими путями и являются пути в блокирующем потоке слоистой сети).

## **2.2. Структуры данных, использованные для эксперимента**

Во входных файлах во всех трёх задачах граф задавался в виде матрицы смежности, которую было удобно хранить в двумерном векторе `std::vector<vector<int>>`.

### **3. Описание плана эксперимента**

#### **3.1. Реализация эксперимента**

Каждый алгоритм был запущен 10 раз на всех входных данных. Значения максимальных потоков записывается в файл output.txt.

### **4. Результаты эксперимента**

#### **4.1. Результаты вычислений**

File name	Answer	File name	Answer	File name	Answer
input_10_0.0.txt	31	input_610_0.0.txt	87	input_1210_0.0.txt	140
input_10_0.5.txt	45	input_610_0.5.txt	16592	input_1210_0.5.txt	31993
input_10_1.0.txt	468	input_610_1.0.txt	33270	input_1210_1.0.txt	65717
input_10_disco.txt	338	input_610_disco.txt	14500	input_1210_disco.txt	30665
input_310_0.0.txt	63	input_910_0.0.txt	150	input_1510_0.0.txt	88
input_310_0.5.txt	7817	input_910_0.5.txt	23495	input_1510_0.5.txt	41844
input_310_1.0.txt	16157	input_910_1.0.txt	49212	input_1510_1.0.txt	80504
input_310_disco.txt	7656	input_910_disco.txt	24059	input_1510_disco.txt	38754
input_1810_0.0.txt	88	input_2110_0.0.txt	122	input_2410_0.0.txt	67
input_1810_0.5.txt	41844	input_2110_0.5.txt	49487	input_2410_0.5.txt	58419
input_1810_1.0.txt	80504	input_2110_1.0.txt	99508	input_2410_1.0.txt	111565
input_1810_disco.txt	38754	input_2110_disco.txt	48794	input_2410_disco.txt	57884
input_2710_0.0.txt	134	input_2710_0.5.txt	74659	input_2710_1.0.txt	145476
input_2710_disco.txt	72513				

#### 4.2. Результаты измерений времени

File name (.txt)	Ford-Fulkerson	Edmonds-Karp	Dinitz
input_10_0.0.txt	127440	10 324	27173
input_10_0.5.txt	16057	15630	9375
input_10_1.0.txt	56586	59710	21465
input_10_disco.txt	18540	30010	23830
input_310_0.0.txt	80273073	4 162 838	15343968
input_310_0.5.txt	8282377259	738123360	82330890
input_310_1.0.txt	9630126909	1261000400	92756710
input_310_disco.txt	2274134051	205158320	68459270
input_610_0.0.txt	232095367	30 876 510	56866514
input_610_0.5.txt	53319666680	4731948700	543455720

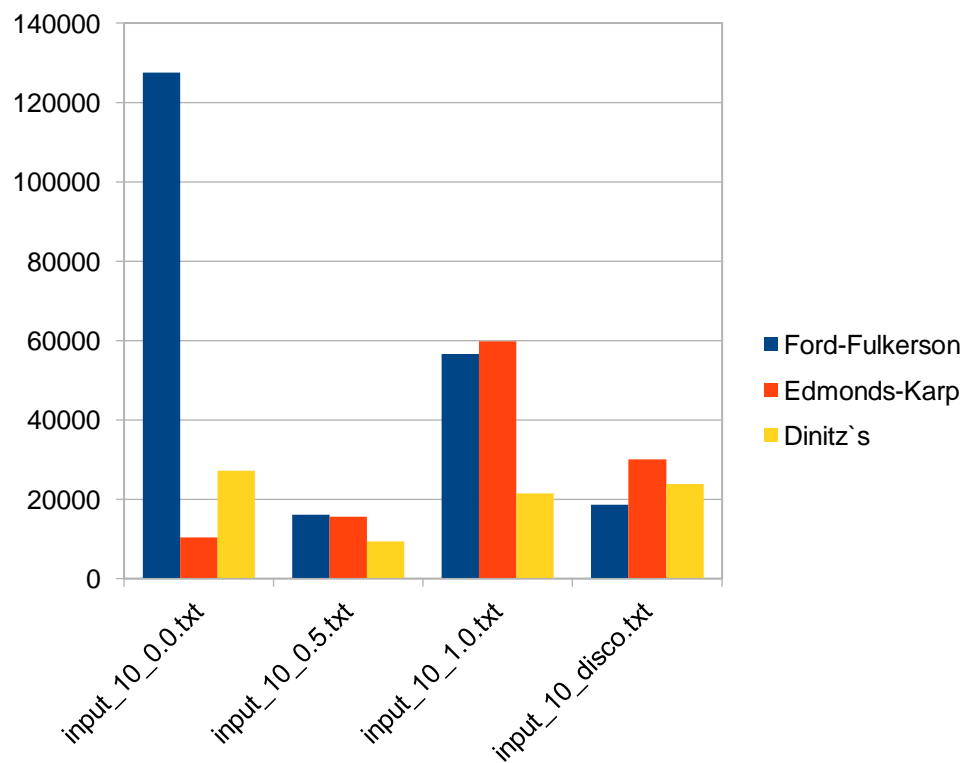
input_610_1.0.txt	1,27655E+11	8787139100	623682453
input_610_disco.txt	19266340235	1509865090	3096208574
input_910_0.0.txt	1599420147	88978176	147476189
input_910_0.5.txt	2,2326E+11	19214461600	202057100
input_910_1.0.txt	3,57073E+11	32962387160	1185641901
input_910_disco.txt	76324853107	5803043910	1225030100
input_1210_0.0.txt	2632324482	154220761	600911381
input_1210_0.5.txt	5,27475E+11	38058140170	4427265960
input_1210_1.0.txt	6,14007E+11	69729595400	1580262550
input_1210_disco.txt	1,99848E+11	12501917420	1783651630

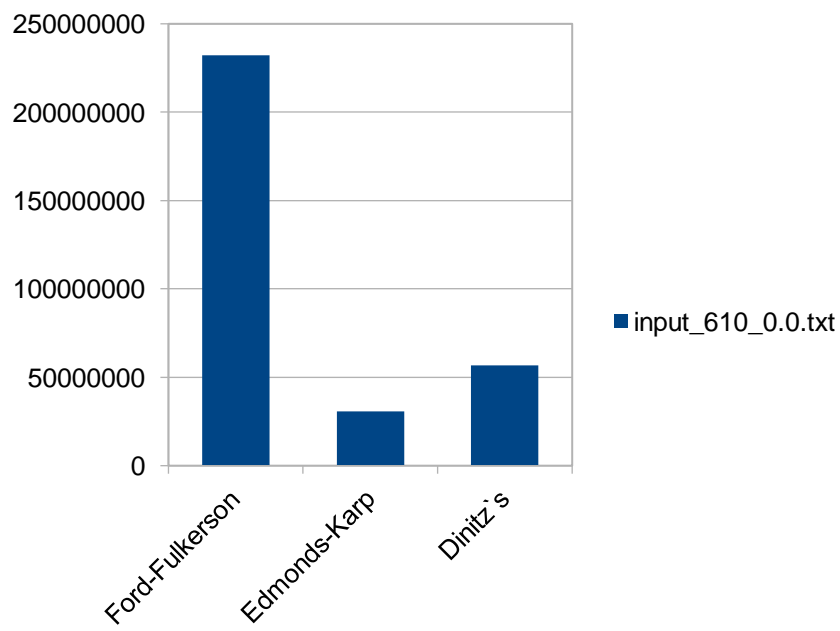
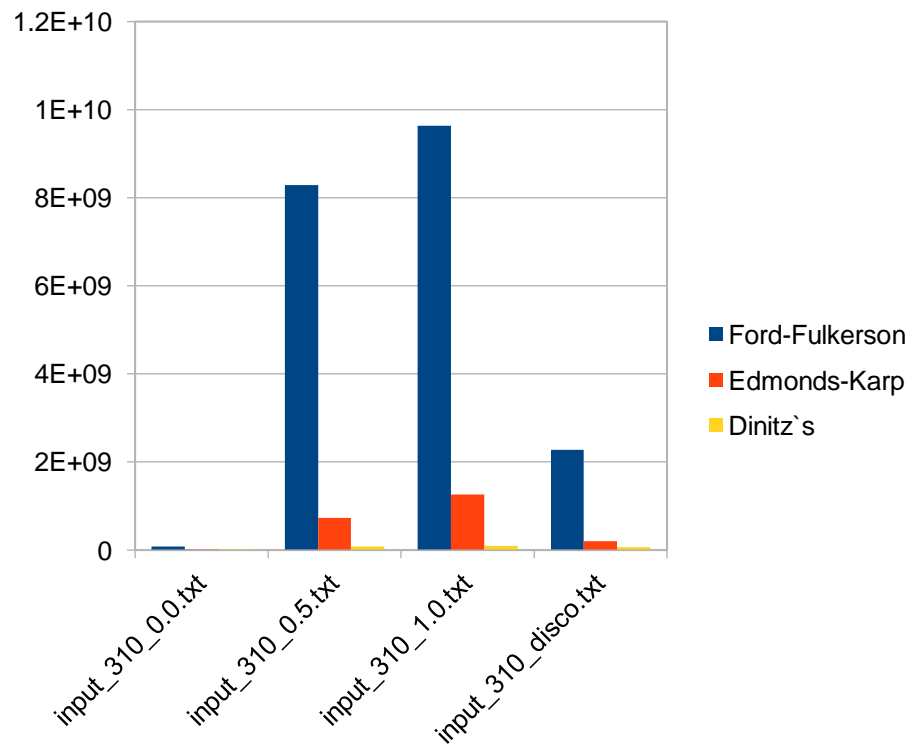
input_1510_0.0.txt	2820678730	149850340	355200804
input_1510_0.5.txt	8,66558E+11	76877987600	7449399770
input_1510_1.0.txt	44126716376	1,33888E+11	2770412010
input_1510_disco.txt	2,71615E+11	24381224600	3265140990
input_1810_0.0.txt	7019550201	184495808	458433700
input_1810_0.5.txt	1,03674E+11	1,51095E+11	10336703760
input_1810_1.0.txt	2,786E+11	2,64992E+11	5427784230
input_1810_disco.txt	7,2878E+11	43964909930	5547541700
input_2110_0.0.txt	2778477657	222254197	658186253
input_2110_0.5.txt	2,47081E+11	1,11009E+11	8797844105

input_2110_1.0.txt	3,5184E+11	1,99709E+11	5358950207
--------------------	------------	-------------	------------

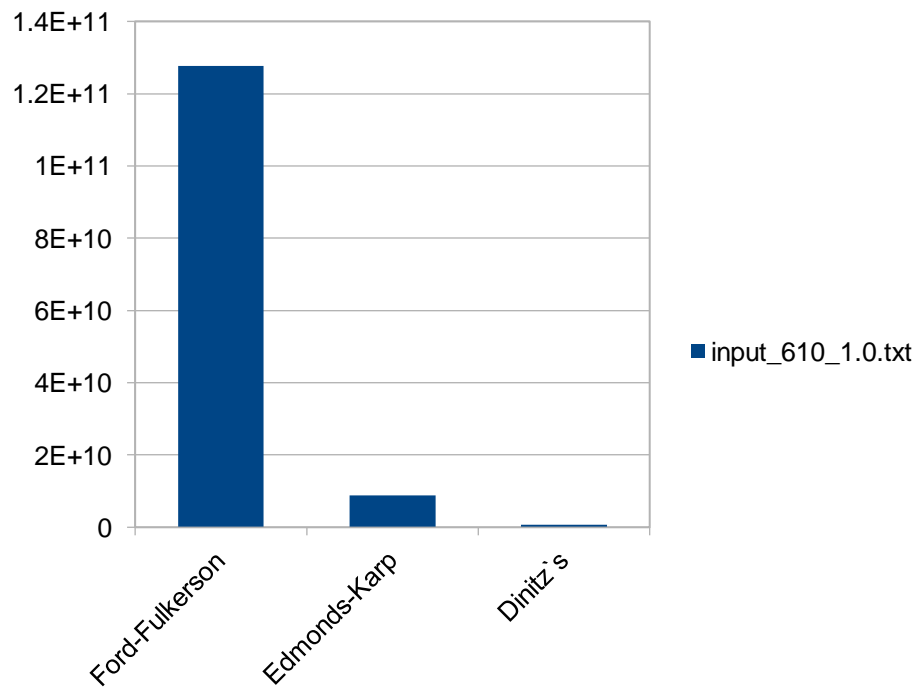
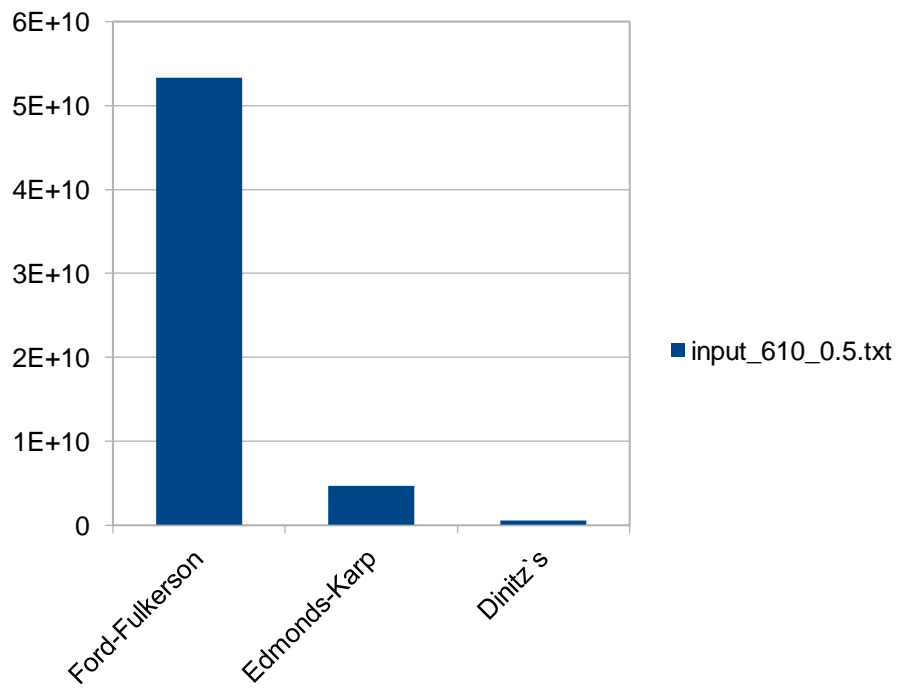
input_2110_disco.txt	67931696812	42646955403	6618573401
input_2410_0.0.txt	2868179016	143078580	539763439
input_2410_0.5.txt	9574798456	1,58479E+11	11182440100
input_2410_1.0.txt	4,79513E+11	3,0168E+11	8177129609
input_2410_disco.txt	1,40875E+11	65195652804	7580890407
input_2710_0.0.txt	12645492424	485365132	1029913419
input_2710_0.5.txt	1,35519E+11	2,4293E+11	19119602605
input_2710_1.0.txt	7,54621E+11	4,44885E+11	11177753907
input_2710_disco.txt	2,53041E+11	95843273201	12844586908

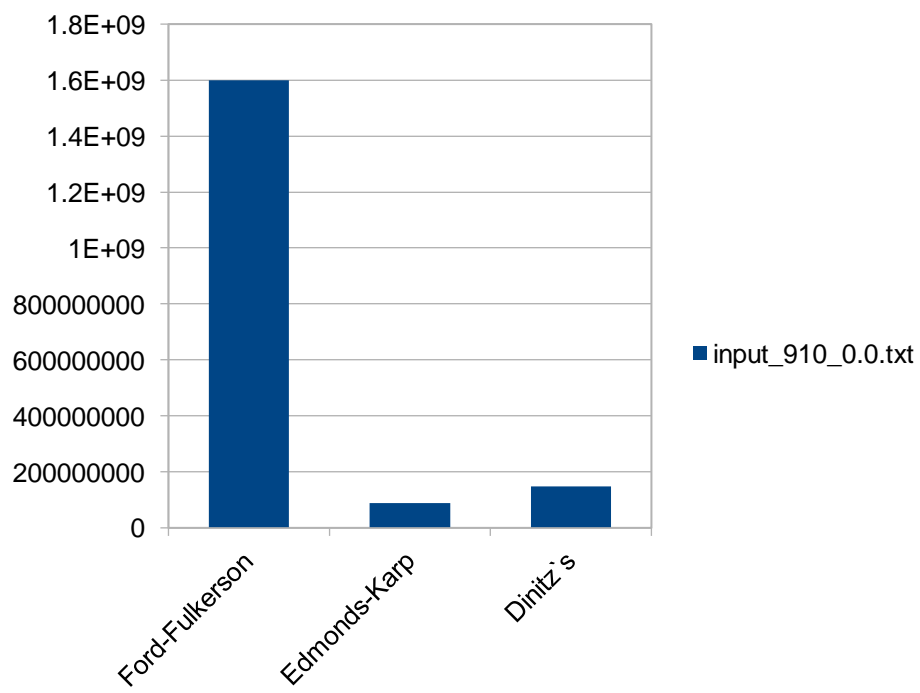
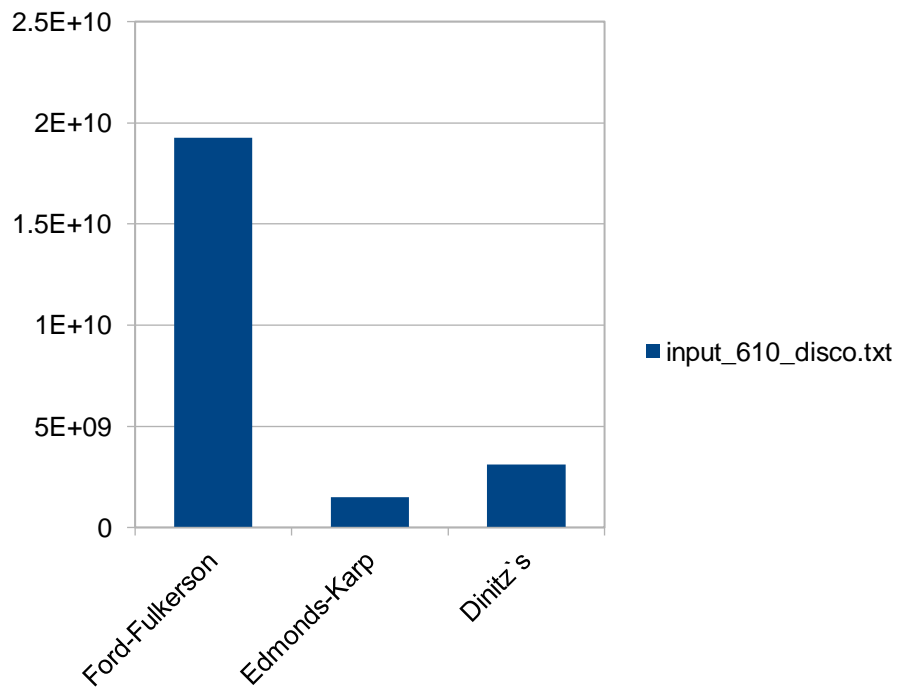
#### 4.3. Графики времени работы разных алгоритмов на одном файле

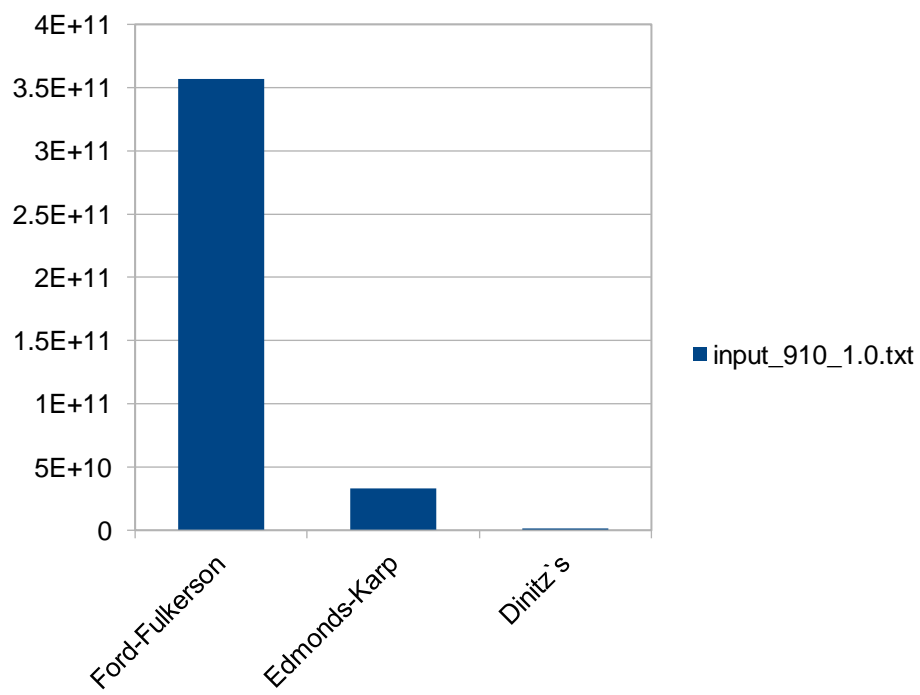
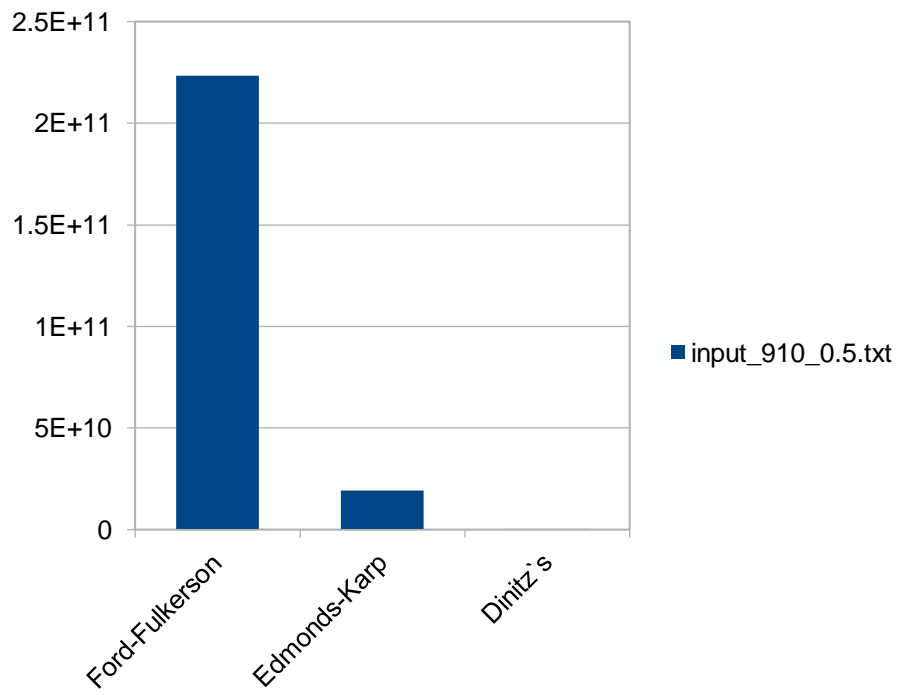


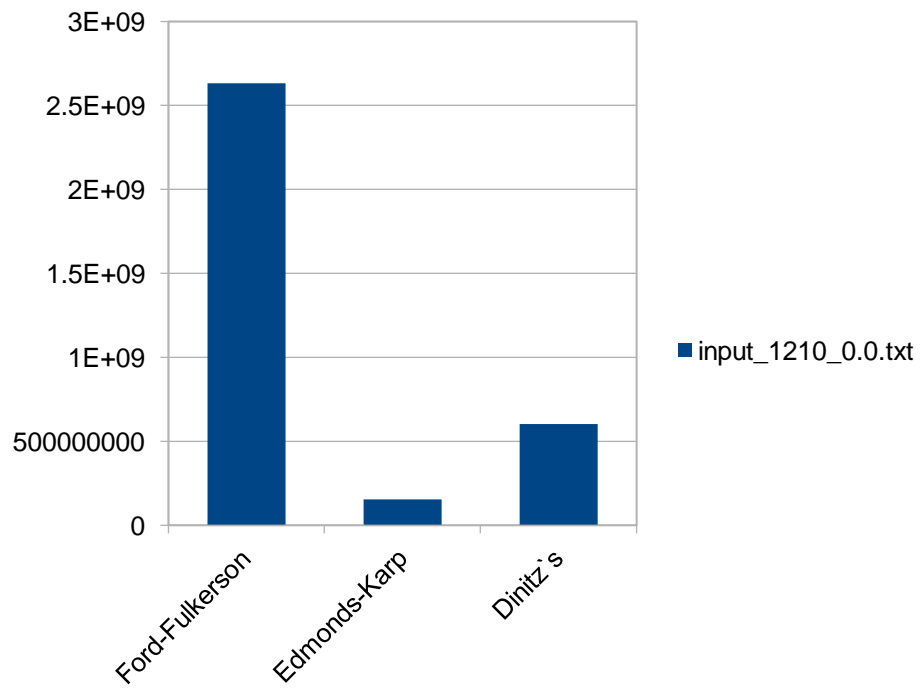
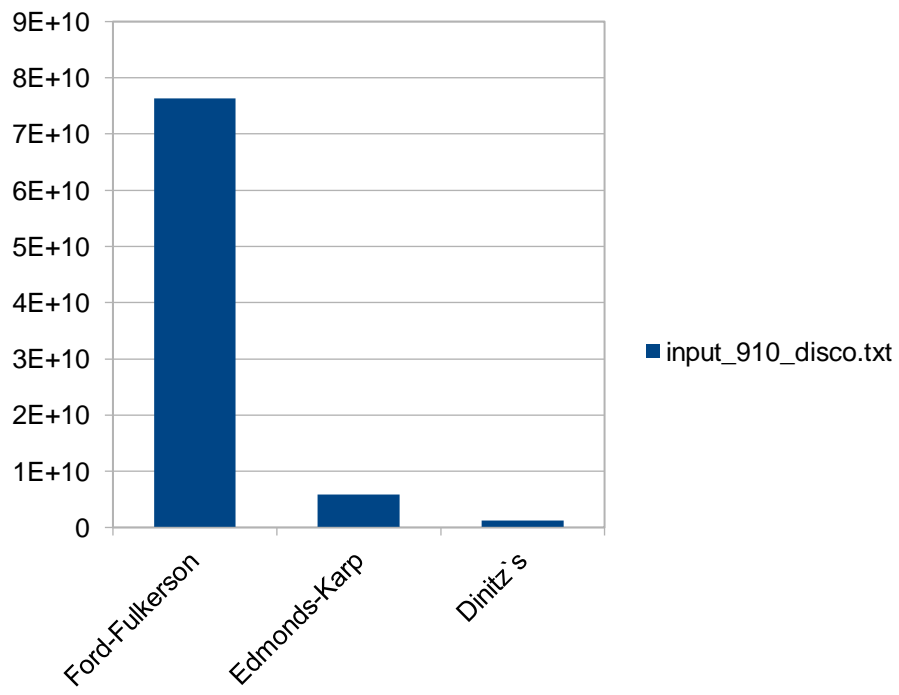


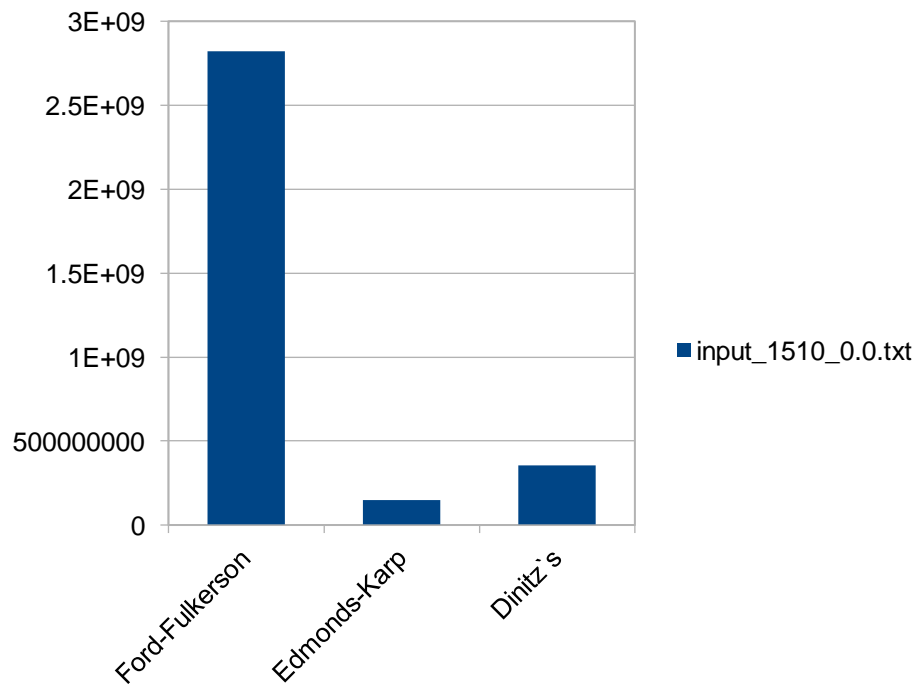
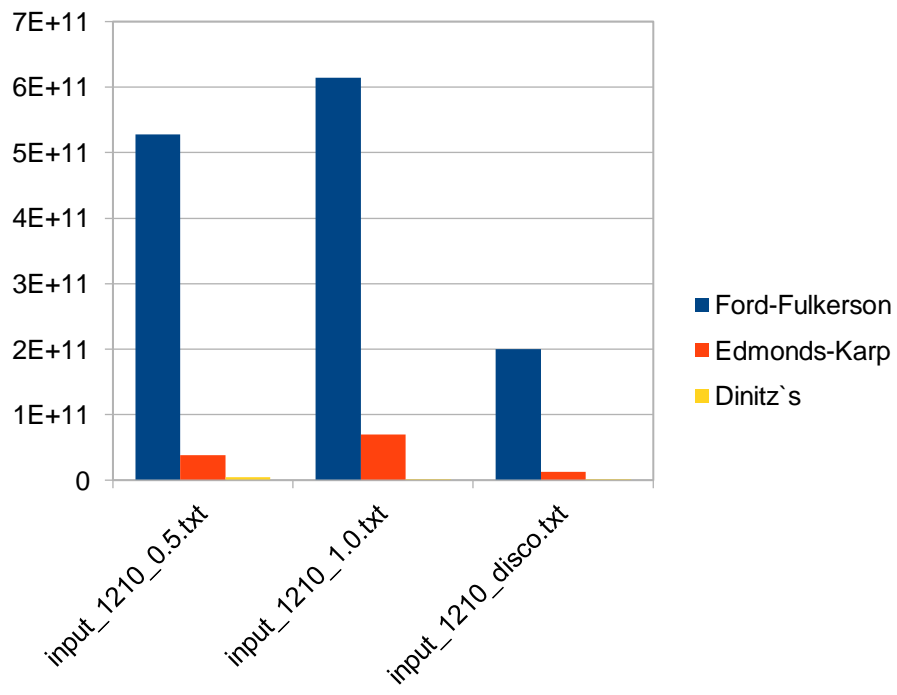


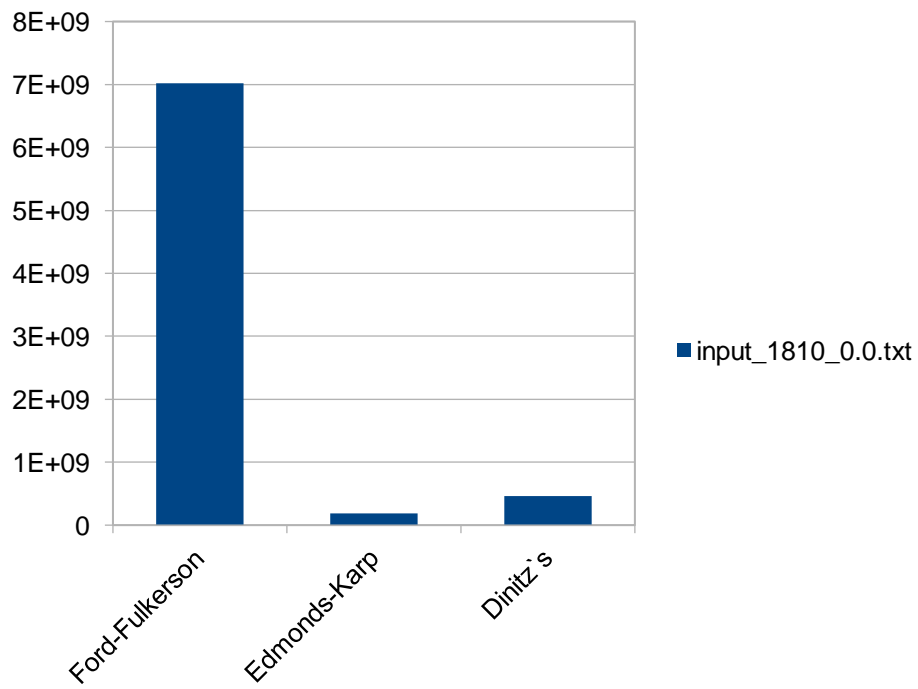
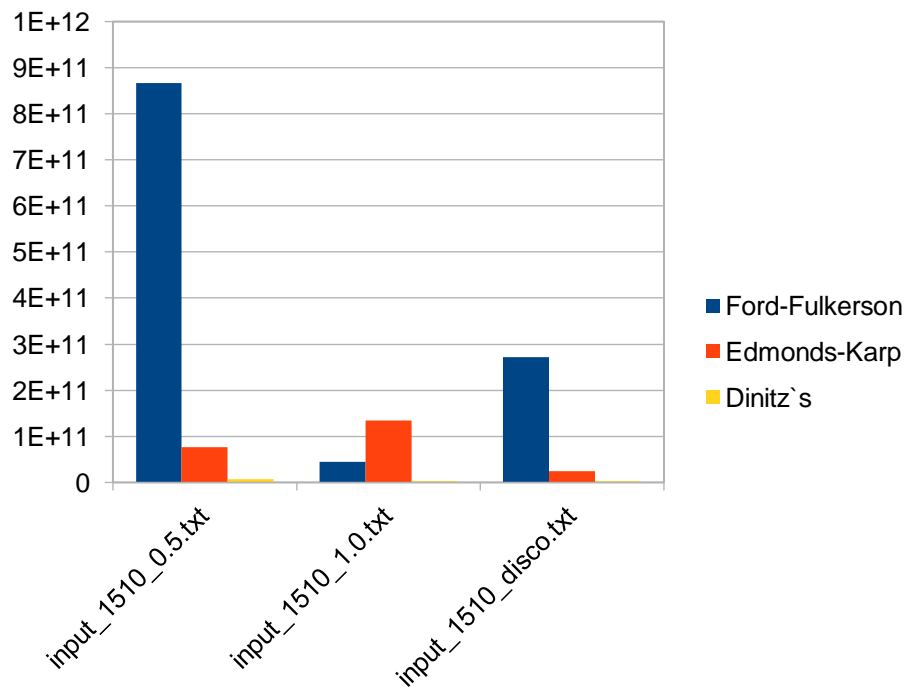


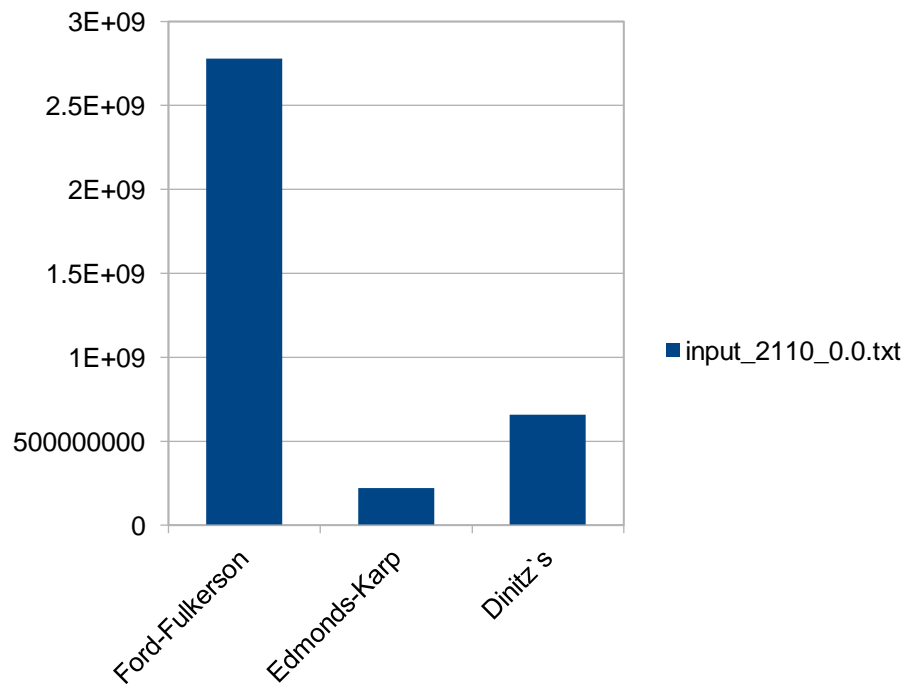
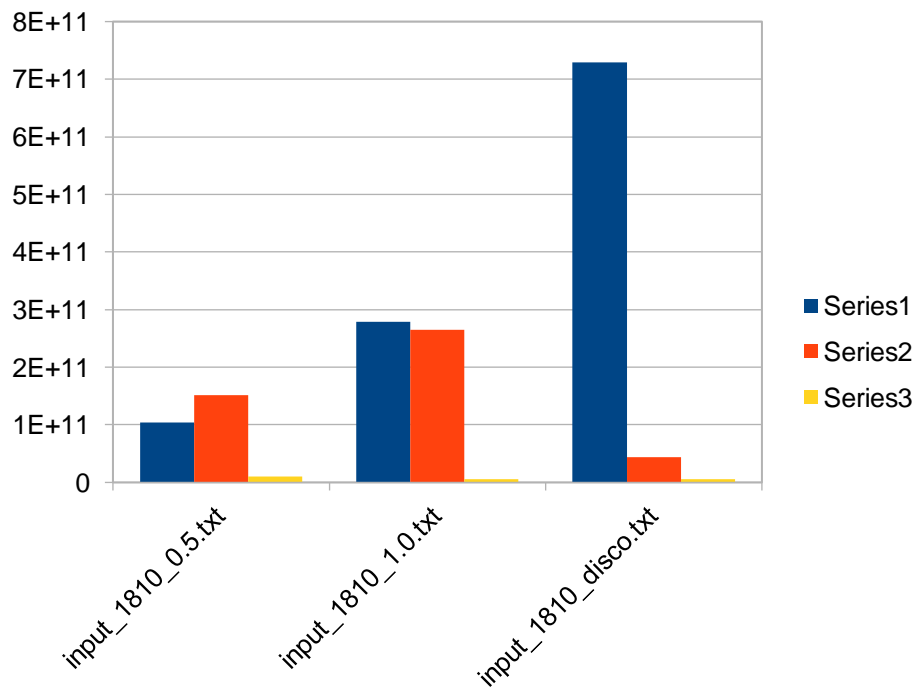


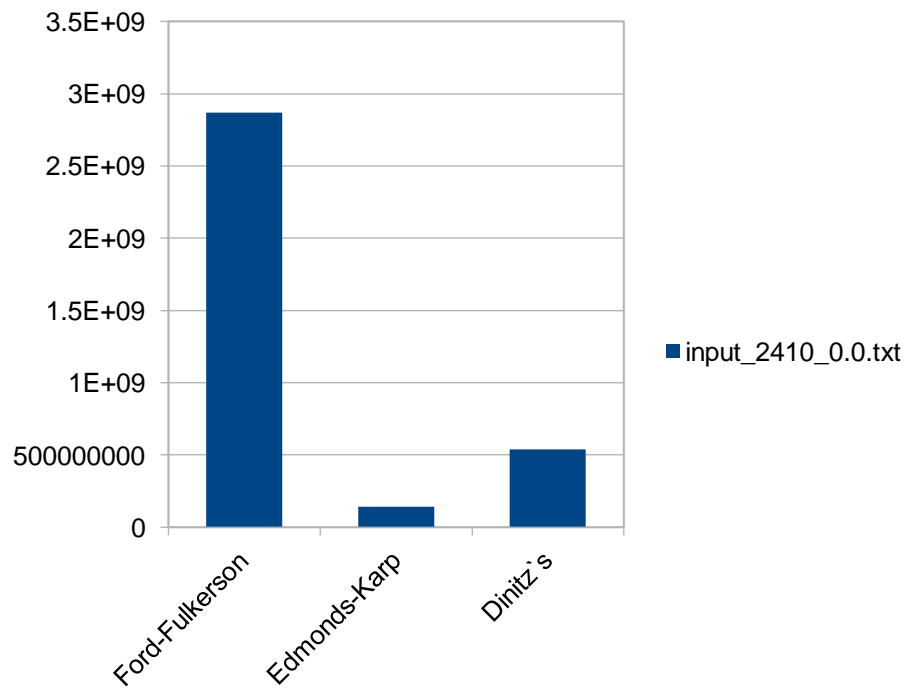
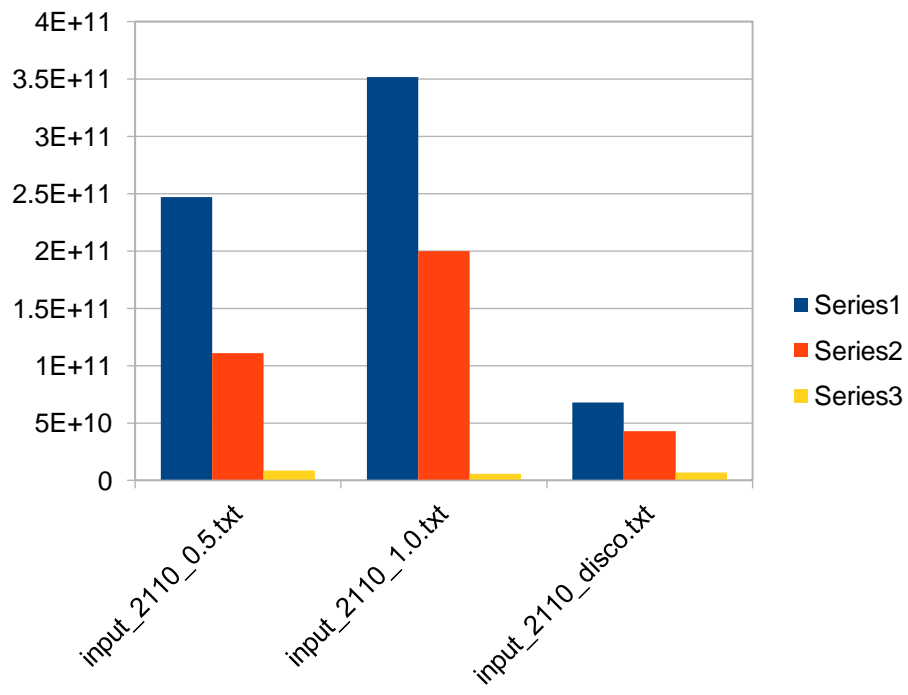




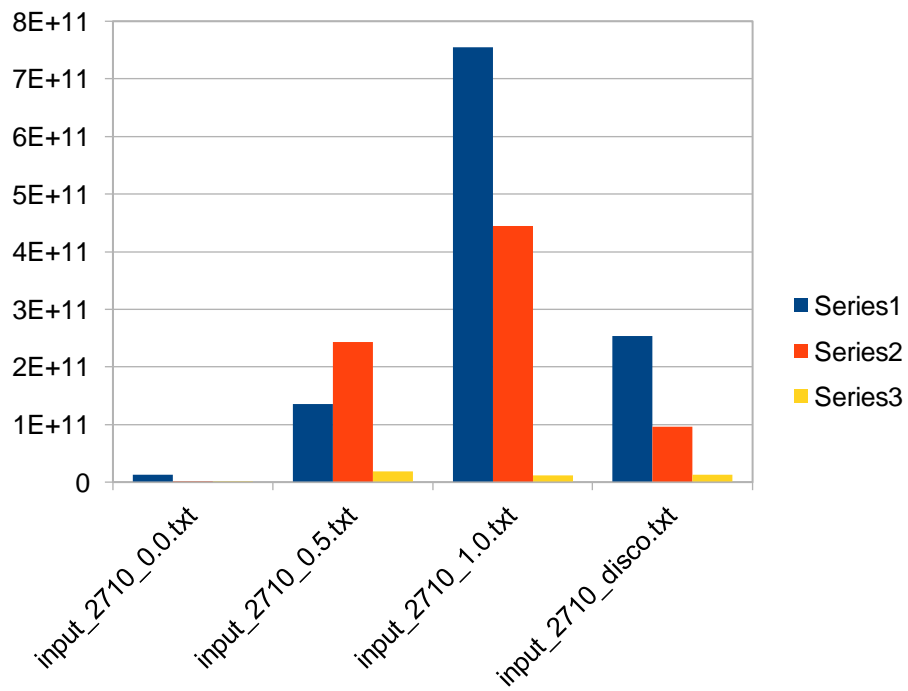
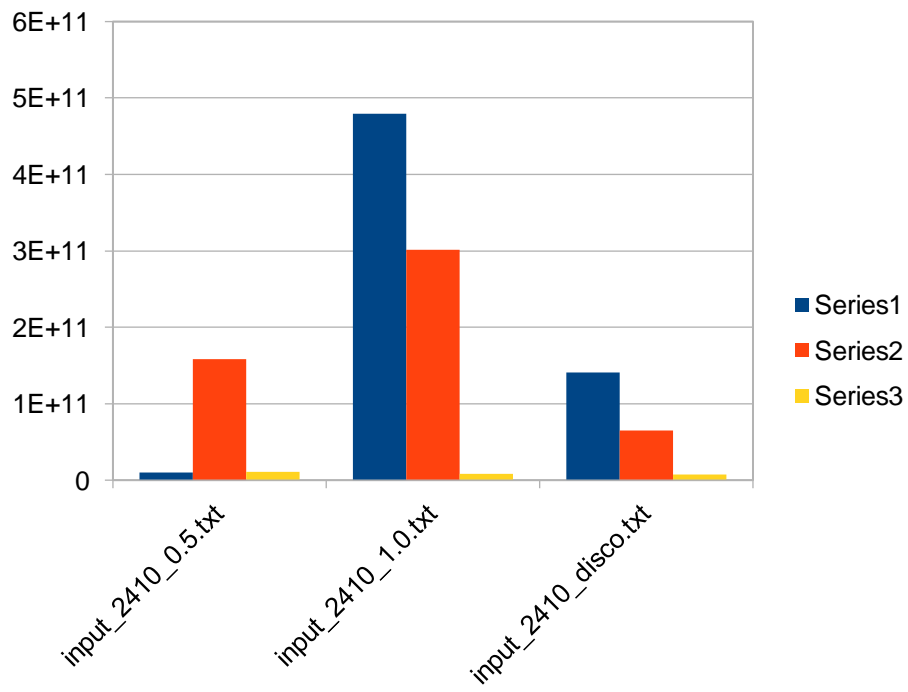




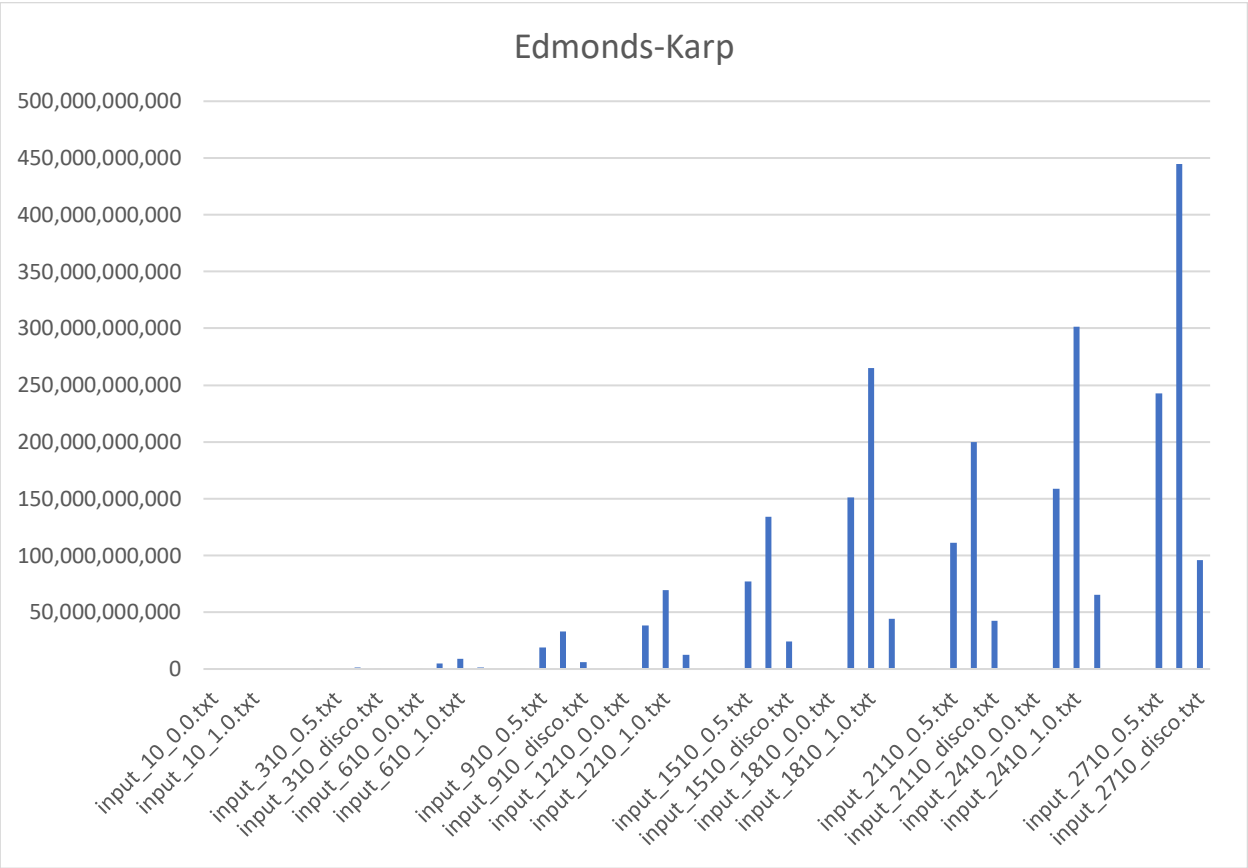
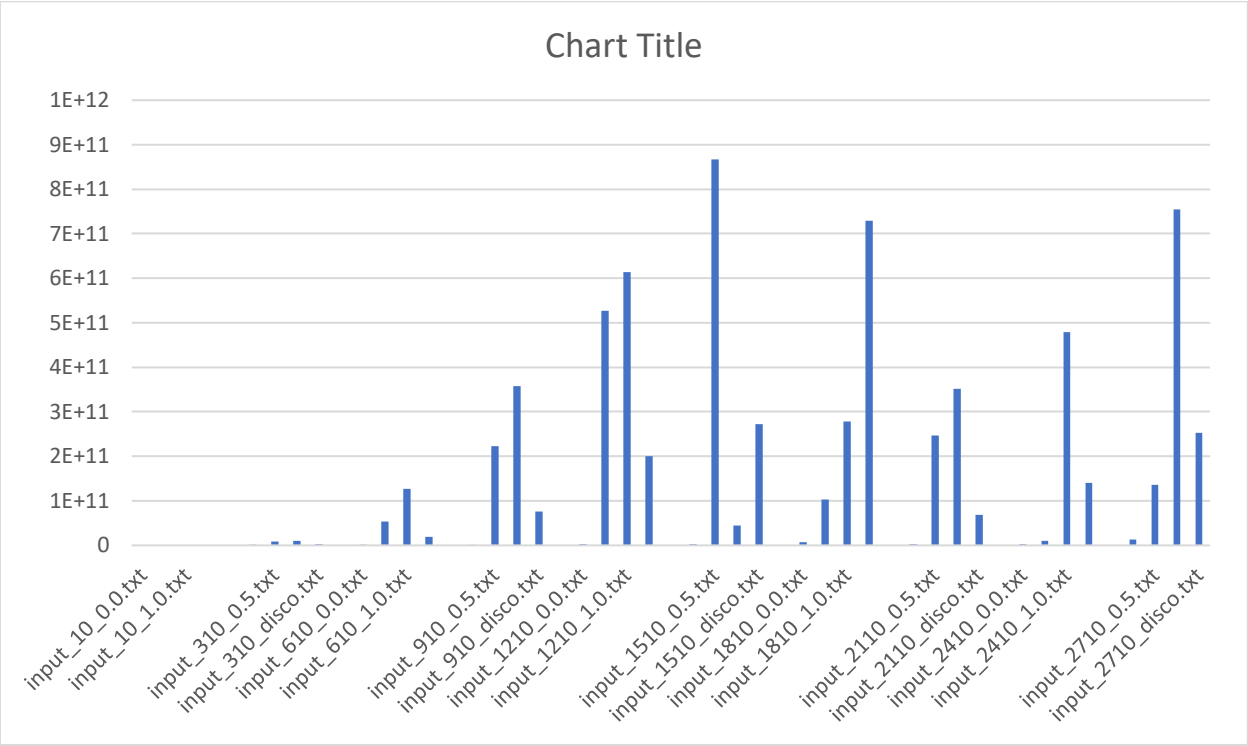


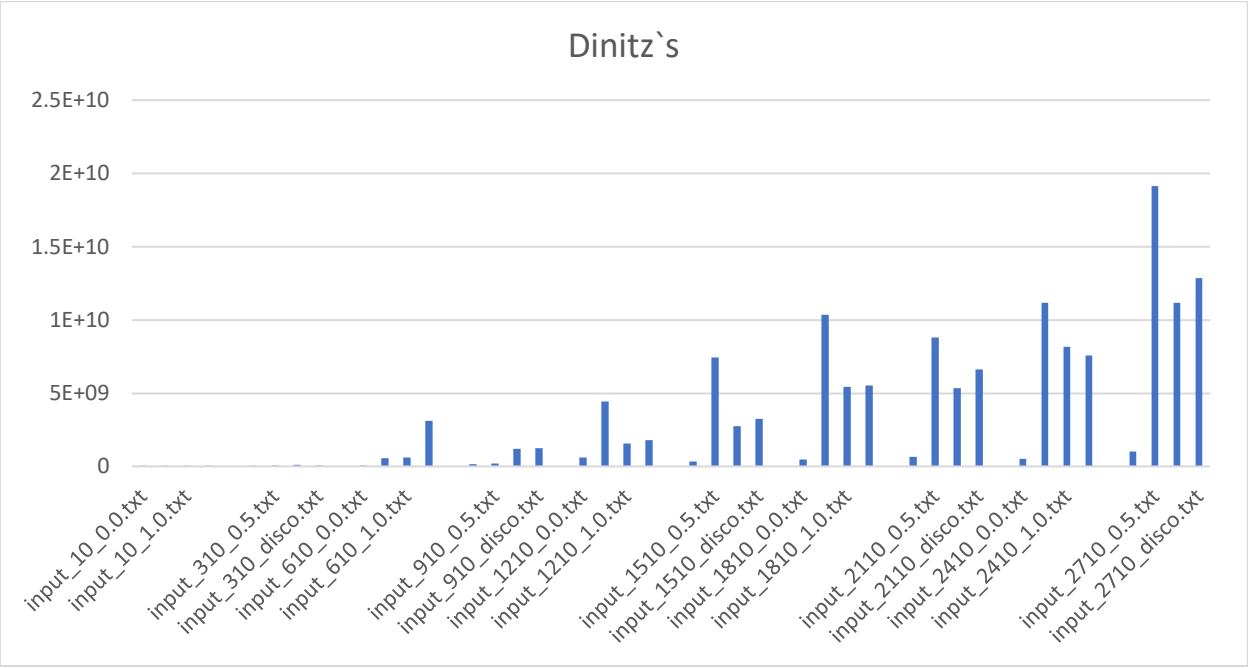






#### 4.4. Графики времени работы одного алгоритма на разных файлах





## **5. Сравнительный анализ алгоритмов**

### **5.1. Асимптотические оценки работы алгоритмов**

Алгоритм Форда-Фалкерсона имеет сложность  $|E| * |f|$ , где  $|f|$  - величина максимального потока.

Алгоритм Эдмондса Карпа находит дополняющий путь за  $O(|E|)$ . Общее число увеличений потока составляет  $O(|E| * |V|)$ . Таким образом, сложность составляет  $O(|V| * |E|^2)$ .

Алгоритм Диница имеет сложность  $O(|E| * |V|^2)$ .

### **5.2. Асимптотические оценки времени работы алгоритмов**

Из данных, приведенных в графиках в пункте 4, можно сделать вывод, что на разреженных графах дольше всего работает алгоритм Форда-Фалкерсона, затем идет алгоритм Эдмонса-Карпа и самый быстрый из них это алгоритм Диница. На графах средней плотности сохраняется тот же порядок, но при больших входных данных алгоритм Форда-Фалкерсона работает в среднем быстрее, чем алгоритм Эдмонса-Карпа. На плотных и несвязных графах алгоритмы работают примерно так же, как и на разреженных. Из графиков видно, что самый быстрый алгоритм – это алгоритм Диница.

## 6. Заключение

В рамках данного контрольного домашнего задания было реализовано 3 алгоритма поиска максимального потока, измерено время выполнения каждого из них на разных входных данных и, на основе полученных данных, были сделаны выводы о каждом из алгоритмов.

## 7. Список литературы

- [illegible]