Struktury Danych i Złożoność Obliczeniowa

Zadanie projektowe nr 2

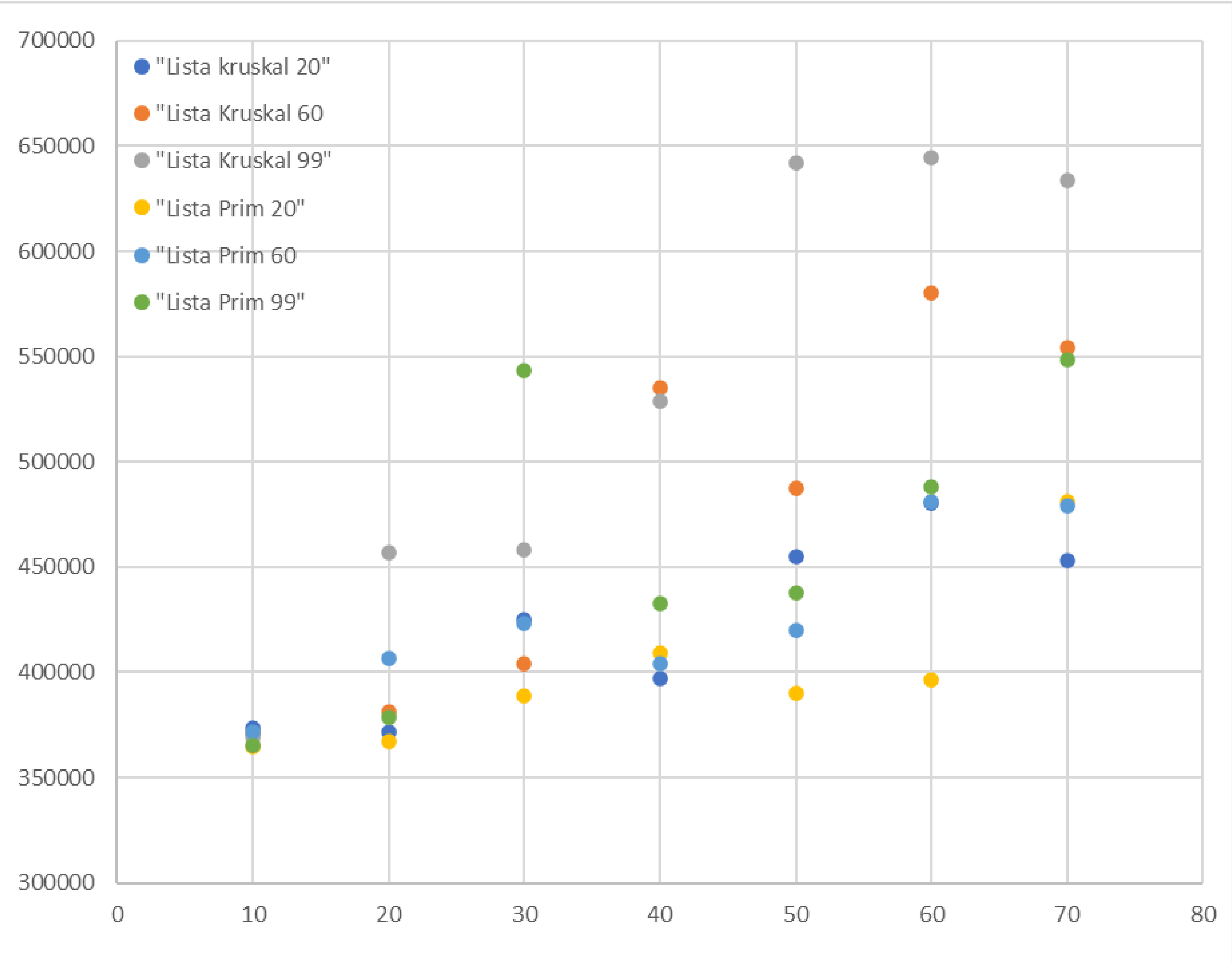
Kamil Wojcieszak

Celem tego projektu jest badanie efektywności algorytmów grafowych w zależności od rozmiaru instancji oraz sposobu reprezentacji grafu w pamięci komputera. Projekt zakłada zaimplementowanie i pomiar czasu działania wybranych algorytmów grafowych, takich jak algorytm Prima i Kruskala do wyznaczania minimalnego drzewa rozpinającego (MST), algorytm Dijkstry i Forda-Bellmana do wyznaczania najkrótszej ścieżki w grafie, oraz algorytm Forda-Fulkersona do wyznaczania maksymalnego przepływu.

Wszystkie czasy na wykresach i tabelkach są w nanosekundach.

Algorytmy MST

Lista

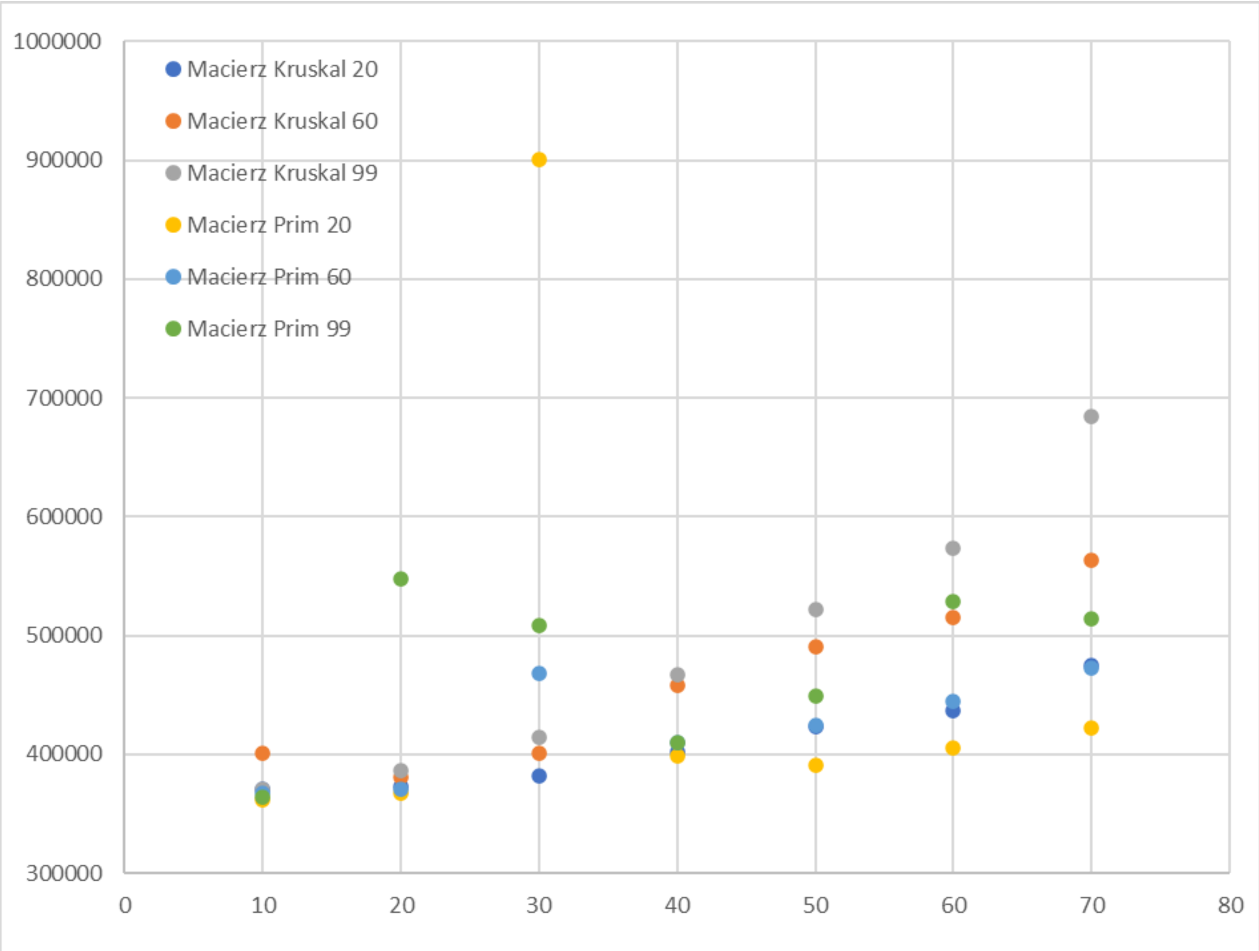


700000
650000
600000
550000
500000
450000
400000
350000
300000
0
10
20
30
40
50
60
70
80
"Lista kruskal 20"
"Lista Kruskal 60
"Lista Kruskal 99"
"Lista Prim 20"
"Lista Prim 60
"Lista Prim 99"

Lista Kruskal	20	60	99
10	373200	370900	368900
20	371800	381000	456700
30	424800	403800	458100
40	397100	535100	528600
50	454900	487500	642200
60	480100	580500	644800
70	453200	554500	633400

Lista Prim	20	60	99
10	364800	371700	365400
20	367000	406300	378900
30	389100	423400	543400
40	409000	404200	432400
50	390000	419800	437600
60	396200	481100	487800
70	481300	479100	548600

Macierz



Macierz Kruskal	20	60	99
10	371000	401300	371000
20	372400	380900	386400
30	382200	400800	414300
40	402400	457600	467200
50	423200	491000	521900

60	437000	514800	573200
70	475000	563400	684400

Macierz Prim	20	60	99
10	362000	367400	364300
20	366800	370800	547700
30	900700	468600	508400
40	398300	410100	409700
50	391100	424500	449100
60	405100	444500	529000
70	422400	472300	514300

Porównując czasy działania algorytmów Kruskala i Prima dla tej samej ilości węzłów w grafie, można zauważyć, że w większości przypadków czasy działania są podobne, ale nie zawsze identyczne.

Czasy działania algorytmu Kruskala dla listy są zazwyczaj nieco większe niż dla macierzy.

Czasy działania algorytmu Prima dla macierzy są zazwyczaj większe niż dla listy.

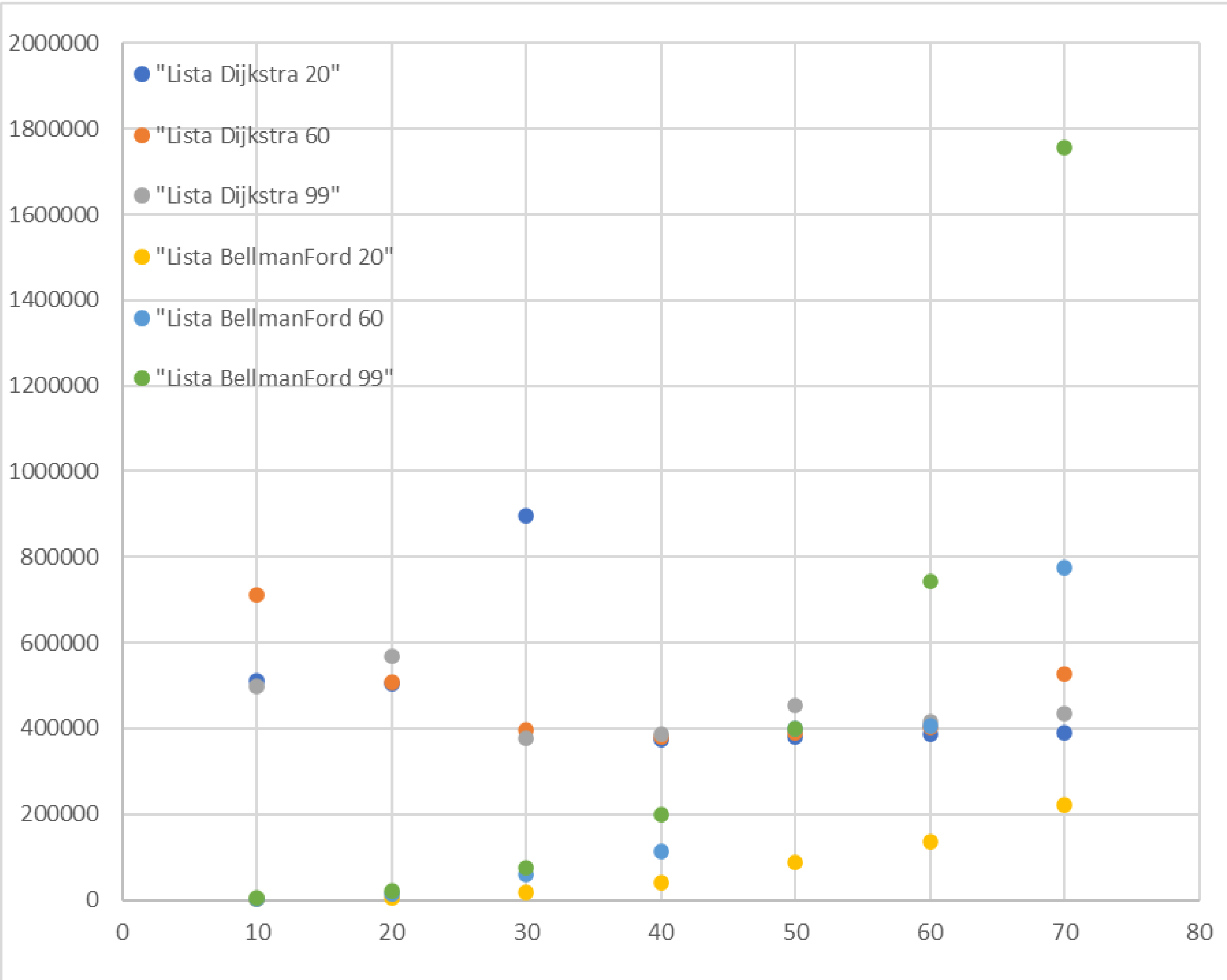
Ogólnie rzecz biorąc, czasy działania obu algorytmów rosną wraz z ilością węzłów w grafie. Jest to zgodne z teorią, ponieważ zarówno algorytm Kruskala, jak i Prima mają złożoność czasową O(E log V), gdzie E to liczba krawędzi, a V to liczba węzłów w grafie.

W niektórych przypadkach można zauważyć nieoczekiwane wyniki, na przykład większy czas działania algorytmu Prima dla mniejszej ilości węzłów w grafie. Może to wynikać z różnych czynników, takich jak sposób implementacji, właściwości grafu (np. gęstość krawędzi) lub wpływ innych czynników zewnętrznych.

Algorytmy

Shortest-path

Lista

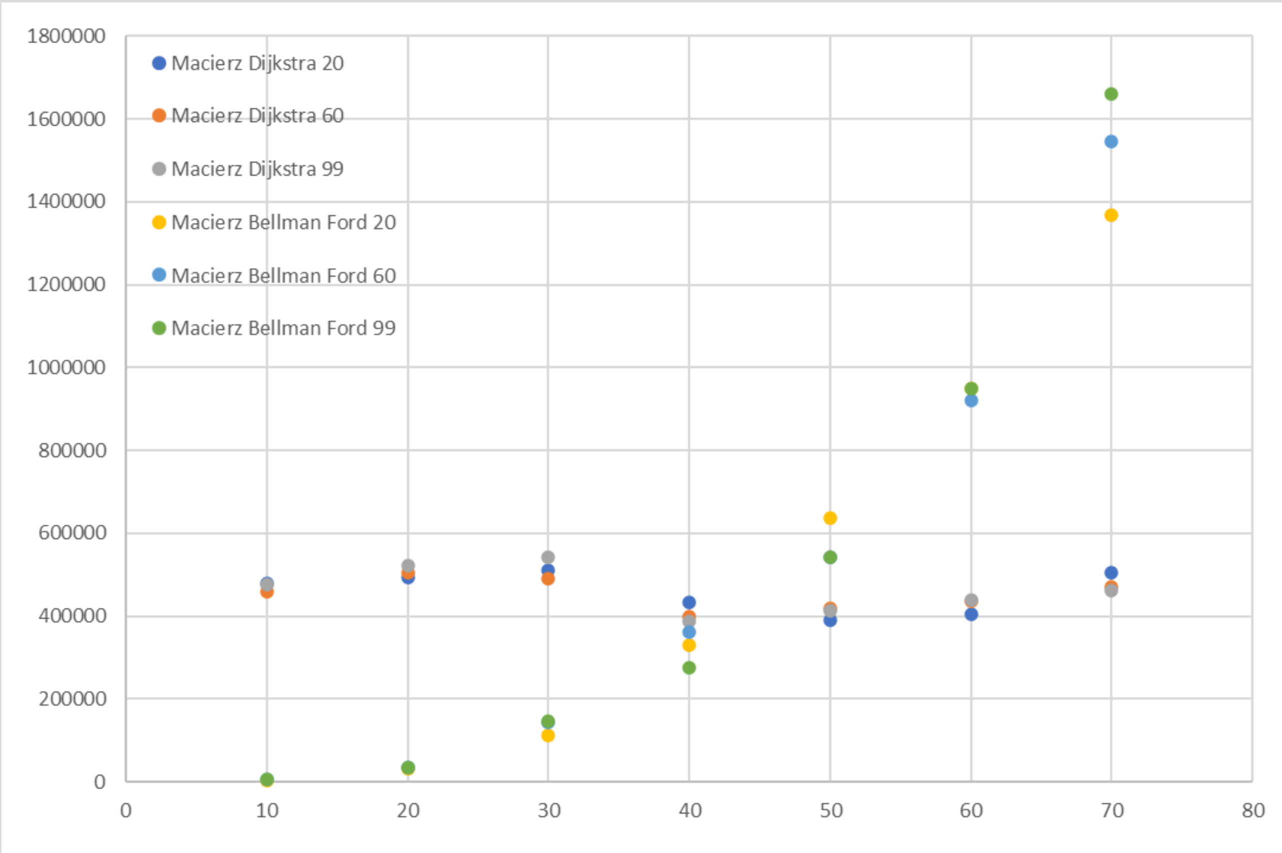


"Lista Dijkstra 20"
"Lista Dijkstra 60
"Lista Dijkstra 99"
"Lista BellmanFord 20"
"Lista BellmanFord 60
"Lista BellmanFord 99"
2000000
1800000
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
0
10
20
30
40
50
60
70
80

Lista Dijkstra	20	60	99
10	512100	711900	499600
20	506100	508400	569500
30	897200	397800	378000
40	373800	381600	387700
50	379100	389700	454100
60	388000	402600	415100
70	391000	528100	434900

Lista Bellman Ford	20	60	99
10	1300	2200	3200
20	6000	14100	21000
30	17700	59600	76000
40	39900	114200	199700
50	86600	399600	400500
60	136800	406200	744900
70	221800	774700	1755200

Macierz



Macierz Dijkstra	20	60	99
10	480000	460500	476900
20	493900	504900	521400
30	510200	491100	542800
40	433100	399200	389000
50	389300	417800	412300
60	404300	436400	437900
70	506200	471300	462700

Macierz Bellman Ford	20	60	99
10	5000	6200	6800
20	33400	35600	35300
30	113100	144600	146200
40	329400	362100	275400
50	637400	541300	541300
60	949300	921300	948400
70	1368500	1545000	1661400

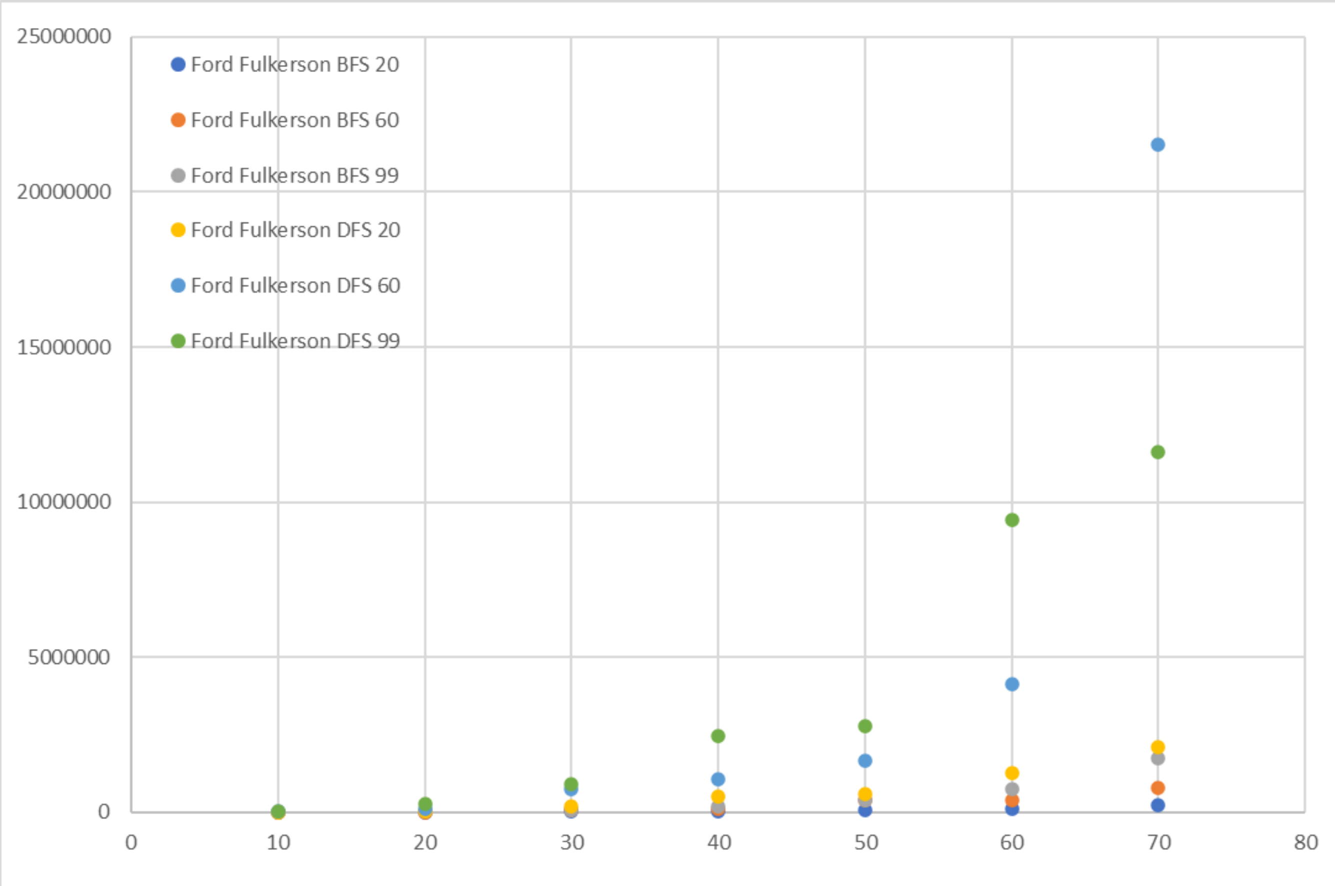
Algorytm Dijkstry:

- Lista: Można zauważyć, że zarówno dla listy, jak i macierzy sąsiedztwa, czas wykonania algorytmu Dijkstry rośnie wraz z wzrostem liczby węzłów w grafie. Te wyniki zgadzają się z teorią, która zakłada złożoność czasową O((V+E)logV), gdzie V to liczba węzłów, a E to liczba krawędzi w grafie. W przypadku algorytmu Dijkstry zaimplementowanego na liście sąsiedztwa, złożoność czasowa wynosi O((V+E)logV), ponieważ wyszukiwanie najmniejszej odległości od wierzchołka źródłowego wymaga operacji w czasie logarytmicznym. Na podstawie wyników można zauważyć, że złożoność czasowa jest zbliżona do teoretycznej zależności.
- Macierz: Wyniki dla macierzy sąsiedztwa również potwierdzają teorię. Czas wykonania rośnie wraz z wzrostem liczby węzłów, co jest zgodne z złożonością czasową O(V^2). Operacje na macierzy sąsiedztwa mają zwykle stały czas wykonania, więc otrzymane wyniki są zbliżone do teoretycznych.

Algorytm Bellmana-Forda:

- Lista: Czas wykonania algorytmu Bellmana-Forda na liście sąsiedztwa rośnie wraz z wzrostem liczby węzłów w grafie. Wyniki te są zgodne z złożonością czasową O(VE), która dotyczy tego algorytmu. Otrzymane czasy wykonania są jednak znacznie większe niż teoretyczna złożoność czasowa. Może to wynikać z różnych czynników, takich jak implementacja algorytmu, szczególne cechy grafu, czy inne czynniki wpływające na czas wykonania.
- Macierz: Wyniki dla macierzy sąsiedztwa również potwierdzają teoretyczną zależność. Czas wykonania rośnie wraz z wzrostem liczby węzłów, co jest zgodne z złożonością czasową O(V^3). Ponownie, otrzymane czasy wykonania są większe niż teoretyczne, co może wynikać z różnych czynników.

Algorytm
Ford Fulkerson



Macierz Ford Fulkerson BFS	20	60	99
10	1300	2200	3200
20	6000	14100	21000
30	17700	59600	76000
40	39900	114200	199700
50	86600	399600	400500
60	136800	406200	744900
70	221800	774700	1755200

Macierz Ford Fulkerson DFS	20	60	99
10	10100	35300	39000
20	25600	132900	258700
30	204900	736100	929000
40	509800	1070200	2479300
50	583900	1666300	2767900
60	1290800	4128200	9421300
70	2105000	21535400	11636200

Ford-Fulkerson

- Algorytm Forda-Fulkersona z BFS: Wyniki pokazują, że czas wykonania algorytmu Forda-Fulkersona z zastosowaniem BFS na macierzy sąsiedztwa rośnie wraz z wzrostem liczby węzłów w grafie. Te wyniki są zgodne z teorią, która mówi, że złożoność czasowa algorytmu Forda-Fulkersona z BFS to O(VE^2), gdzie V to liczba węzłów, a E to liczba krawędzi w grafie. Otrzymane czasy wykonania są zbliżone do teoretycznych zależności, ale mogą się różnić ze względu na różne czynniki, takie jak implementacja i charakterystyka grafu.
- Algorytm Forda-Fulkersona z DFS: Wyniki dla algorytmu Forda-Fulkersona z zastosowaniem DFS na macierzy sąsiedztwa również potwierdzają teoretyczne zależności. Czas wykonania rośnie wraz z wzrostem liczby węzłów w grafie, co jest zgodne z złożonością czasową O(V(E+V)), która dotyczy tego algorytmu. Otrzymane czasy wykonania są zbliżone do teoretycznych, ale mogą się różnić w zależności od różnych czynników.