
Zadanie projektowe nr 2

Badanie efektywności algorytmów grafowych w zależności od rozmiaru instancji oraz sposobu reprezentacji grafu w pamięci komputera

Autor sprawozdania: Michał Dziedziak, 263901

Nazwisko i Imię prowadzącego kurs: Mgr. inż. Antoni Sterna

Dzień i godzina zajęć: Wtorek, 15:15 - 16:55 TN

Spis treści

1	Wstęp	3
1.1	Zarys projektu	3
1.2	Algorytmy	3
1.2.1	Zaimplementowane algorytmy	3
1.2.2	Teoretyczne złożoności obliczeniowe	3
2	Plan eksperymentu	3
2.1	Założenia	3
2.2	Generowanie grafu	4
2.3	Pomiar czasu	4
2.4	Losowanie populacji	4
3	Dane określające Wyniki pomiarów	5
3.1	Algorytmy znajdowania minimalnego drzewa rozpinającego	5
3.1.1	Algorytm Prima i macierz sąsiedztwa	5
3.1.2	Algorytm Prima i lista sąsiedztwa	6
3.1.3	Algorytm Kruskala i macierz sąsiedztwa	7
3.1.4	Algorytm Kruskala i lista sąsiedztwa	8
3.2	Algorytmy znajdowania najkrótszej ścieżki w grafie	9
3.2.1	Algorytm Dijkstry i macierz sąsiedztwa	9
3.2.2	Algorytm Dijkstry i lista sąsiedztwa	10
3.2.3	Algorytm Bellmana-Forda i macierz sąsiedztwa	11
3.2.4	Algorytm Bellmana-Forda i lista sąsiedztwa	12
4	Wykresy	12
5	Wnioski	12
5.1	Przykładowe wyliczenia	12
5.2	Komentarz	13

Spis tabel

1	Pomiary algorytmu Prima na macierzy sąsiedztwa	5
2	Pomiary algorytmu Prima na liście sąsiedztwa	6
3	Pomiary algorytmu Kruskala na macierzy sąsiedztwa	7
4	Pomiary algorytmu Kruskala na liście sąsiedztwa	8
5	Pomiary algorytmu Dijkstry na macierzy sąsiedztwa	9
6	Pomiary algorytmu Dijkstry na liście sąsiedztwa	10
7	Pomiary algorytmu Bellmana-Forda na macierzy sąsiedztwa	11
8	Pomiary algorytmu Bellmana-Forda na liście sąsiedztwa	12

1 Wstęp

1.1 Zarys projektu

W ramach projektu w języku C++ zaimplementowane zostały wybrane algorytmy znajdowania najmniejszego drzewa rozpinającego i najkrótszej ścieżki w grafie. Do reprezentacji grafów użyta została macierz sąsiedztw i lista sąsiedztw. Poniższe sprawozdanie bada zależność czasu od ilości wierzchołków i gęstości grafu podczas wykonywania poszczególnych algorytmów.

1.2 Algorytmy

1.2.1 Zaimplementowane algorytmy

- Najmniejsze drzewo rozpinające
 - Algorytm Prima
 - Algorytm Kruskala
- Najkrótsza ścieżka w grafie
 - Algorytm Dijkstry
 - Algorytm Bellmana-Forda

1.2.2 Teoretyczne złożoności obliczeniowe

W tej sekcji przedstawię teoretyczne złożoności obliczeniowe operacji na grafach, celem późniejszego porównania ich z wynikami testów.

Algorytm	Reprezentacja grafu	Pesymistyczna złożoność	Średnia złożoność	Optymistyczna złożoność
Prima	Macierz sąsiedztwa	$O(V^2 \log V)$	$O((V + E) \log V)$	$O(E \log V)$
Prima	Lista sąsiedztwa	$O((V + E) \log V)$	$O((V + E) \log V)$	$O((V + E) \log V)$
Kruskal	Macierz sąsiedztwa	$O(E \log E)$	$O(E \log E)$	$O(E \log E)$
Kruskal	Lista sąsiedztwa	$O(E \log E)$	$O(E \log E)$	$O(E \log E)$
Dijkstra	Macierz sąsiedztwa	$O(V^2 \log V)$	$O(V^2 \log V)$	$O(V^2 \log V)$
Dijkstra	Lista sąsiedztwa	$O((E + V) \log V)$	$O((E + V) \log V)$	$O(E + V \log V)$
Bellman-Ford	Macierz sąsiedztwa	$O(V^2)$	$O(V^2)$	$O(V^2)$
Bellman-Ford	Lista sąsiedztwa	$O(V^2)$	$O(V^2)$	$O(V^2)$

2 Plan eksperymentu

2.1 Założenia

- Macierz sąsiedztwa i niektóre algorytmy wykorzystują umowną wartość maksymalnego int'a do reprezentowania braku krawędzi pomiędzy wierzchołkami lub nieskończonego kosztu dojścia do danego wierzchołka.
- Wagi krawędzi są liczbami całkowitymi.
- Testy wykonywane są dla grafów posiadających od 50 do 500 wierzchołków (z krokiem równym 50). Dla każdej liczby wierzchołków wykonywane są cztery testy dla gęstości grafu równej kolejno 25%, 50%, 75%, 99%. Test dla każdej kombinacji ilości wierzchołków i gęstości grafu wykonywany jest sto razy i ostateczny czas jest średnią tych czasów.

2.2 Generowanie grafu

Dla każdego testu generowany jest nowy graf. Algorytm analizuje każde potencjalne połączenie pomiędzy wierzchołkami i ustawia je lub nie (ma na to 50% szans). Kolejno waga takiego połączenia losowana jest z zakresu od 0 lub -1000 (w zależności od algorytmu) do 1000. Algorytm losuje czy dodaje nowe połączenie do momentu, kiedy wszystkie pozostałe połączenia muszą zostać stworzone w celu zapewnienia wymaganej gęstości (wtedy tworzy pozostałe wierzchołki z losowymi wagami).

2.3 Pomiar czasu

Do pomiaru czasu została napisana osobna klasa "Timer". Korzysta ona z funkcji QueryPerformanceCounter i QueryPerformanceFrequency umożliwiających pomiar czasu z dokładnością do mikro sekund. Klasa ta określa upływ czasu, bazując na dokładnym liczniku.

2.4 Losowanie populacji

Do losowania populacji również została napisana osobna klasa RandomGenerator. Umożliwia ona losowanie pojedynczych liczb lub całych zbiorów. Korzysta z mt19937 - generatora liczb pseudolosowych.

3 Dane określające Wyniki pomiarów

3.1 Algorytmy znajdowania minimalnego drzewa rozpinającego

3.1.1 Algorytm Prima i macierz sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.303281
50	0.750000	0.402145
50	0.990000	0.470321
100	0.250000	0.758356
100	0.500000	1.268732
100	0.750000	1.937749
100	0.990000	2.432933
150	0.250000	1.920544
150	0.500000	3.140806
150	0.750000	4.315116
150	0.990000	5.366154
200	0.250000	2.688177
200	0.500000	4.421315
200	0.750000	6.163672
200	0.990000	7.866423
250	0.250000	4.142686
250	0.500000	8.937206
250	0.750000	19.523090
250	0.990000	15.169887
300	0.250000	6.771716
300	0.500000	10.155652
300	0.750000	15.480223
300	0.990000	23.588072
350	0.250000	8.250819
350	0.500000	16.226921
350	0.750000	23.091856
350	0.990000	31.765903
400	0.250000	12.179870
400	0.500000	20.330405
400	0.750000	33.550595
400	0.990000	40.603692
450	0.250000	17.468599
450	0.500000	31.035094
450	0.750000	43.067905
450	0.990000	53.132712
500	0.250000	21.477893
500	0.500000	36.477335
500	0.750000	51.247519
500	0.990000	66.563587

Tabela 1: Pomiary algorytmu Prima na macierzy sąsiedztwa

3.1.2 Algorytm Prima i lista sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.257786
50	0.750000	0.368533
50	0.990000	0.467485
100	0.250000	0.606198
100	0.500000	1.045913
100	0.750000	1.600941
100	0.990000	1.923094
150	0.250000	1.379166
150	0.500000	2.276298
150	0.750000	3.336044
150	0.990000	4.360509
200	0.250000	2.922207
200	0.500000	5.400914
200	0.750000	6.607999
200	0.990000	9.318399
250	0.250000	3.626270
250	0.500000	6.393082
250	0.750000	10.775849
250	0.990000	15.068256
300	0.250000	6.548412
300	0.500000	11.568113
300	0.750000	45.353180
300	0.990000	34.131315
350	0.250000	9.333016
350	0.500000	20.843599
350	0.750000	33.070768
350	0.990000	42.511576
400	0.250000	14.562317
400	0.500000	29.197952
400	0.750000	43.727784
400	0.990000	56.887612
450	0.250000	19.438359
450	0.500000	37.071992
450	0.750000	56.111448
450	0.990000	76.871865
500	0.250000	30.430238
500	0.500000	46.319270
500	0.750000	70.187380
500	0.990000	82.945406

Tabela 2: Pomiary algorytmu Prima na liście sąsiedztwa

3.1.3 Algorytm Kruskala i macierz sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.221651
50	0.750000	0.273010
50	0.990000	0.302807
100	0.250000	0.603217
100	0.500000	0.889653
100	0.750000	1.155309
100	0.990000	1.298558
150	0.250000	1.404983
150	0.500000	1.932349
150	0.750000	2.322570
150	0.990000	2.822679
200	0.250000	2.532478
200	0.500000	3.570259
200	0.750000	4.428116
200	0.990000	5.231842
250	0.250000	5.767008
250	0.500000	7.143056
250	0.750000	7.491960
250	0.990000	10.828563
300	0.250000	5.284594
300	0.500000	7.372174
300	0.750000	7.950661
300	0.990000	10.300609
350	0.250000	9.264268
350	0.500000	13.994538
350	0.750000	15.508083
350	0.990000	15.924039
400	0.250000	9.885155
400	0.500000	13.844131
400	0.750000	15.855709
400	0.990000	20.524412
450	0.250000	11.205335
450	0.500000	16.287721
450	0.750000	21.388701
450	0.990000	26.052395
500	0.250000	16.034102
500	0.500000	22.998907
500	0.750000	26.305869
500	0.990000	33.972631

Tabela 3: Pomiary algorytmu Kruskala na macierzy sąsiedztwa

3.1.4 Algorytm Kruskala i lista sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.300723
50	0.750000	0.434206
50	0.990000	0.571312
100	0.250000	0.666572
100	0.500000	1.211527
100	0.750000	1.801261
100	0.990000	2.106896
150	0.250000	1.200869
150	0.500000	2.214547
150	0.750000	3.242499
150	0.990000	4.199985
200	0.250000	2.081641
200	0.500000	4.325247
200	0.750000	5.889981
200	0.990000	7.921957
250	0.250000	3.272140
250	0.500000	6.201776
250	0.750000	9.876468
250	0.990000	13.550605
300	0.250000	5.876288
300	0.500000	10.127750
300	0.750000	16.097142
300	0.990000	21.109249
350	0.250000	7.244957
350	0.500000	15.078331
350	0.750000	22.933416
350	0.990000	29.698925
400	0.250000	10.197249
400	0.500000	20.287461
400	0.750000	30.602796
400	0.990000	40.395956
450	0.250000	13.448449
450	0.500000	28.389262
450	0.750000	38.761766
450	0.990000	50.918679
500	0.250000	15.671779
500	0.500000	34.591204
500	0.750000	48.842685
500	0.990000	65.437668

Tabela 4: Pomiary algorytmu Kruskala na liście sąsiedztwa

3.2 Algorytmy znajdowania najkrótszej ścieżki w grafie

3.2.1 Algorytm Dijkstry i macierz sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.049796
50	0.750000	0.044260
50	0.990000	0.038507
100	0.250000	0.136127
100	0.500000	0.166586
100	0.750000	0.149942
100	0.990000	0.123939
150	0.250000	0.279066
150	0.500000	0.365823
150	0.750000	0.301460
150	0.990000	0.246690
200	0.250000	0.459391
200	0.500000	0.586763
200	0.750000	0.507006
200	0.990000	0.414611
250	0.250000	0.703869
250	0.500000	0.924586
250	0.750000	0.768380
250	0.990000	0.631720
300	0.250000	1.024951
300	0.500000	1.312713
300	0.750000	1.381106
300	0.990000	1.097974
350	0.250000	1.700746
350	0.500000	1.736259
350	0.750000	1.419579
350	0.990000	1.147681
400	0.250000	1.709360
400	0.500000	2.468254
400	0.750000	2.351288
400	0.990000	1.855333
450	0.250000	2.236671
450	0.500000	2.815838
450	0.750000	2.454902
450	0.990000	2.301735
500	0.250000	2.967897
500	0.500000	3.479210
500	0.750000	3.427004
500	0.990000	2.316485

Tabela 5: Pomiary algorytmu Dijkstry na macierzy sąsiedztwa

3.2.2 Algorytm Dijkstry i lista sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.029328
50	0.750000	0.036269
50	0.990000	0.040040
100	0.250000	0.062150
100	0.500000	0.086850
100	0.750000	0.107739
100	0.990000	0.136985
150	0.250000	0.117788
150	0.500000	0.176553
150	0.750000	0.244567
150	0.990000	0.283340
200	0.250000	0.161951
200	0.500000	0.272885
200	0.750000	0.416856
200	0.990000	0.657230
250	0.250000	0.302850
250	0.500000	0.625786
250	0.750000	0.875380
250	0.990000	1.137031
300	0.250000	0.471892
300	0.500000	0.724115
300	0.750000	0.915850
300	0.990000	1.169388
350	0.250000	0.634547
350	0.500000	1.075721
350	0.750000	1.575341
350	0.990000	1.585468
400	0.250000	0.507708
400	0.500000	1.269795
400	0.750000	1.916187
400	0.990000	2.294401
450	0.250000	0.795218
450	0.500000	1.446324
450	0.750000	2.170751
450	0.990000	2.689522
500	0.250000	0.824275
500	0.500000	1.610873
500	0.750000	2.449523
500	0.990000	3.202803

Tabela 6: Pomiary algorytmu Dijkstry na liście sąsiedztwa

3.2.3 Algorytm Bellmana-Forda i macierz sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	1.874670
50	0.750000	1.788075
50	0.990000	1.826481
100	0.250000	12.014201
100	0.500000	15.322946
100	0.750000	12.838924
100	0.990000	12.218119
150	0.250000	40.326488
150	0.500000	58.125953
150	0.750000	51.329406
150	0.990000	49.416849
200	0.250000	93.855294
200	0.500000	133.497982
200	0.750000	115.526071
200	0.990000	112.336478
250	0.250000	173.120152
250	0.500000	322.795843
250	0.750000	321.959591
250	0.990000	267.248087
300	0.250000	432.904448
300	0.500000	610.496616
300	0.750000	635.184549
300	0.990000	411.033466
350	0.250000	507.204401
350	0.500000	756.598331
350	0.750000	669.522109
350	0.990000	588.094800
400	0.250000	733.877673
400	0.500000	1083.603165
400	0.750000	1014.887489
400	0.990000	903.750843
450	0.250000	1059.969061
450	0.500000	1542.740936
450	0.750000	1412.401980
450	0.990000	1277.851132
500	0.250000	1474.593716
500	0.500000	2113.087035
500	0.750000	1936.552625
500	0.990000	1712.972877

Tabela 7: Pomiary algorytmu Bellmana-Forda na macierzy sąsiedztwa

3.2.4 Algorytm Bellmana-Forda i lista sąsiedztwa

Liczba wierzchołków	Gęstość	Czas [ms]
50	0.500000	0.557555
50	0.750000	0.727694
50	0.990000	0.921011
100	0.250000	1.813399
100	0.500000	4.040461
100	0.750000	6.560684
100	0.990000	9.204846
150	0.250000	6.975900
150	0.500000	16.146445
150	0.750000	28.208296
150	0.990000	37.435486
200	0.250000	19.906129
200	0.500000	42.176963
200	0.750000	70.390731
200	0.990000	90.969152
250	0.250000	42.201848
250	0.500000	82.780494
250	0.750000	134.779592
250	0.990000	175.659666
300	0.250000	73.661505
300	0.500000	144.722326
300	0.750000	223.686539
300	0.990000	302.092524
350	0.250000	116.090112
350	0.500000	237.738000
350	0.750000	358.780572
350	0.990000	491.716774
400	0.250000	171.901412
400	0.500000	358.598470
400	0.750000	542.735733
400	0.990000	769.730502
450	0.250000	245.497919
450	0.500000	511.182012
450	0.750000	830.811591
450	0.990000	1149.816387
500	0.250000	337.645842
500	0.500000	778.188315
500	0.750000	1201.477696
500	0.990000	1600.392871

Tabela 8: Pomiary algorytmu Bellmana-Forda na liście sąsiedztwa

4 Wykresy

5 Wnioski

5.1 Przykładowe wyliczenia

W tej sekcji porównane zostaną czasy otrzymane eksperymentalnie z tymi wynikającymi z teorii. Porównywane będą zależności czasu wykonania danych operacji dla rozmiarów:

5.2 Komentarz