

AOD Lista 3

Mateusz Gancarz

11 czerwca 2023

1 Algorytm Dijkstry - wersja podstawowa

1.1 Opis implementacji algorytmu

Zwykła implementacja Dijkstry polega na utworzeniu kolejki priorytetowej dla wierzchołków, gdzie ich dystans od źródła będzie decydował o ich priorytecie. Na początku wrzucamy źródło z odległością 0, a następnie wrzucamy każdego jego sąsiada, gdy jeszcze nie był odwiedzony oraz odległość do tego wierzchołka jest mniejsza niż wcześniejsza zarejestrowana, z powiększonym dystansem o wagę krawędzi prowadzącą do niego. Przy każdej następnej iteracji pobieramy pierwszy wierzchołek z kolejki priorytetowej oraz powtarzamy poprzednie czynności aż nie dojdziemy do pożądanego wierzchołka.

Po zakończeniu działania algorytmu, zwracany jest wektor $dist$, który zawiera najkrótsze odległości od źródła do wszystkich wierzchołków w grafie.

1.2 Złożoność obliczeniowa

Algorytm Dijkstry odwiedza każdy wierzchołek dokładnie raz ($O(V)$) i sprawdza każdy wierzchołek sąsiedni po krawędziach ($O(E)$). Każde dodanie wierzchołka do kolejki priorytetowej ma złożoność $O(\log V)$. Więc złożoność Dijkstry z kolejką priorytetową jest równa $O(E \log V + V)$.

2 Algorytm Diala

2.1 Opis implementacji algorytmu

Algorytm Diala polega na stworzeniu listy kubełków, do których trafiać będą wierzchołki o dystansie przynależącym do danego kubełka, gdzie maksymalna liczba kubełków wynosić będzie $V * C$. Algorytm zaczynamy od wrzucenia źródła do kubełka z dystansem 0, a następnie dodajemy do odpowiednich kubełków jego sąsiadów z dystansem powiększonym o wagę krawędzi. Przy każdej iteracji pobieramy pierwszy element z pierwszego napotkanego niepustego kubełka, przy czym indeks aktualnego kubełka cały czas porusza się do przodu z każdą iteracją. Algorytm kończymy jak dojdziemy do pożądanego wierzchołka.

2.2 Złożoność obliczeniowa

Algorytm Diala ma złożoność czasową $O(E + VC)$. Drugi składnik sumy bierze się stąd, że maksymalna liczba kubełków może być równa VC w przypadku kiedy każda krawędź miałaby maksymalną wagę. Jest ona zależna od maksymalnej odległości w grafie, co może prowadzić do znacznego zmniejszenia liczby iteracji w porównaniu z klasycznym algorytmem Dijkstry. Dzięki grupowaniu wierzchołków o podobnych odległościach w kubełkach, algorytm Dial przetwarza tylko te kubełki, których odległość jest mniejsza niż aktualnie badana odległość, co może znacznie przyspieszyć działanie algorytmu dla grafów o dużym rozmiarze lub z dużą liczbą krawędzi.

3 Algorytm Radix Heap

3.1 Opis implementacji algorytmu

Implementacja jest podobna do zwykłej implementacji Dijkstry, z tą różnicą że zamiast kolejki priorytetowej używamy struktury Radix Heap. Polega ona na umieszczaniu elementów w odpowiednich kubełkach, ponumerowanych od 0 do 31, gdzie na przykład do kubełka o numerze n trafiają wierzchołki, do których dystans mieści się w zakresie 2^n do $2^{n+1} - 1$. Za pomocą funkcji pop możemy pobrać pierwszy wierzchołek ze struktury, czyli pierwszy element z pierwszego niepustego kubełka.

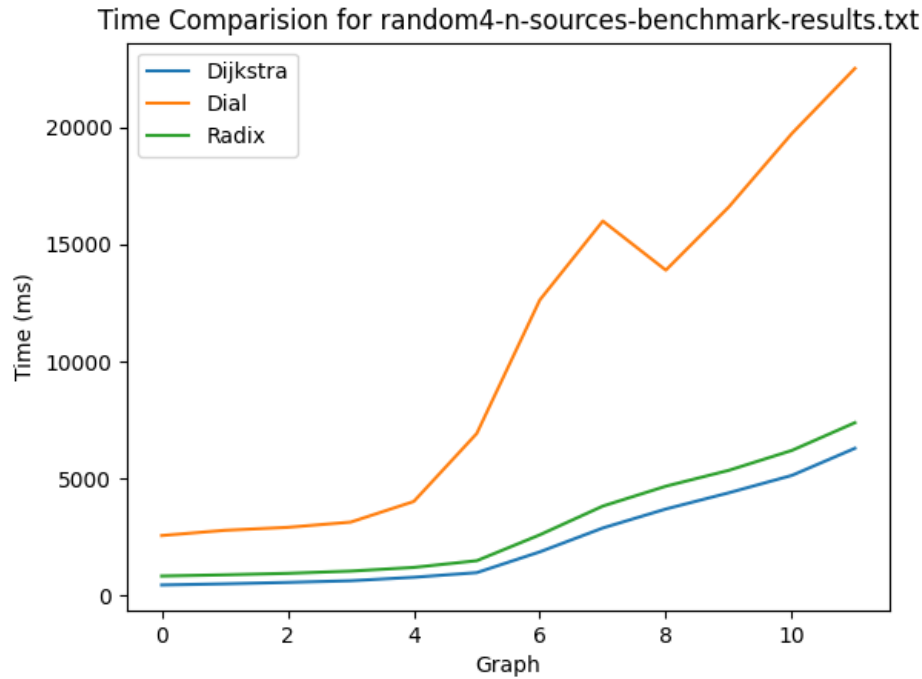
3.2 Złożoność czasowa

Radix Heap przechowuje elementy w kolejce opartej na ich priorytetach i bitach ich wartości. Dzięki temu umożliwia wydajniejsze porównywanie i wstawianie elementów. Załóżmy, że mamy E krawędzi i V wierzchołków w grafie. Algorytm Dijkstry odwiedza każdą krawędź co najwyżej raz, stąd składa się z E operacji. Oprócz tego, dla każdego wierzchołka algorytm wykonuje operację wstawiania do struktury Radix Heap, która ma złożoność $O(\log VC)$, ponieważ największy możliwy dystans jest równy VC , a wrzucanie elementów do struktury polega na znalezieniu najbliższej potęgi 2. Z tego wychodzi, że złożoność algorytmu jest równa $O(E + V \log VC)$

4 Testy

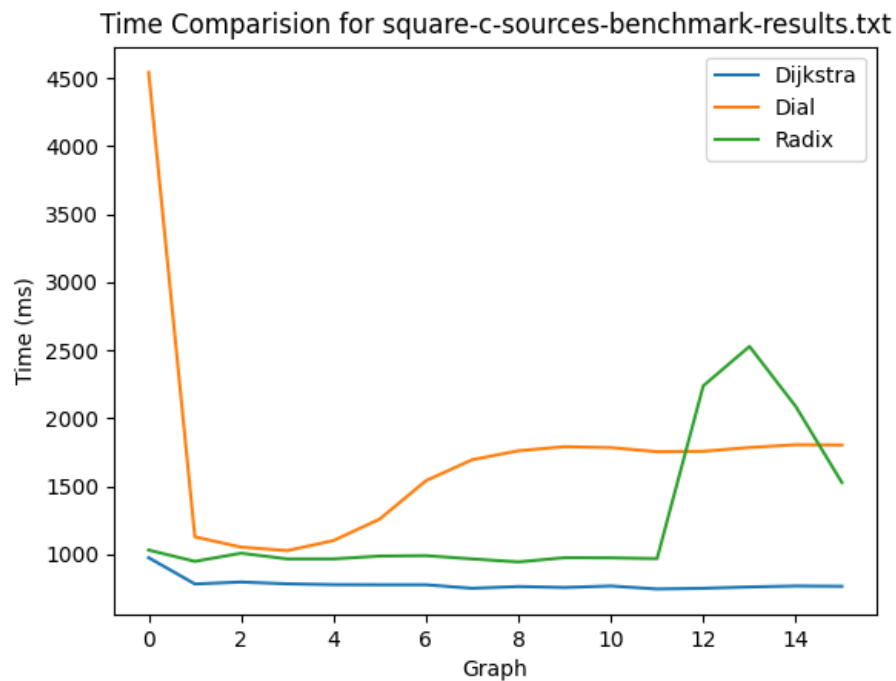
n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
1024	4096	1024	440	0.0134277	2554	0.0779419	823	0.025116
2048	8192	2048	489	0.0298462	2778	0.169556	874	0.0533447
4096	16384	4096	550	0.0671387	2905	0.354614	938	0.114502
8192	32768	8192	621	0.151611	3129	0.763916	1034	0.252441
16384	65536	16384	770	0.375977	4011	1.9585	1191	0.581543
32768	131072	32768	968	0.945312	6920	6.75781	1478	1.44336
65536	262144	65536	1856	3.625	12629	24.666	2585	5.04883
131072	524288	131072	2878	11.2422	16005	62.5195	3814	14.8984
262144	1048576	262143	3689	28.8203	13905	108.633	4664	36.4375
524288	2097152	524288	4383	68.4844	16610	259.531	5344	83.5
1048576	4194304	1048576	5127	160.219	19749	617.156	6195	193.594
2097152	8388608	2097152	6286	392.875	22534	1408.38	7381	461.312

Tabela 1: Porównanie średnich czasów obliczania odległości od źródła do wszystkich wierzchołków w grafie dla rodziny Random4-n



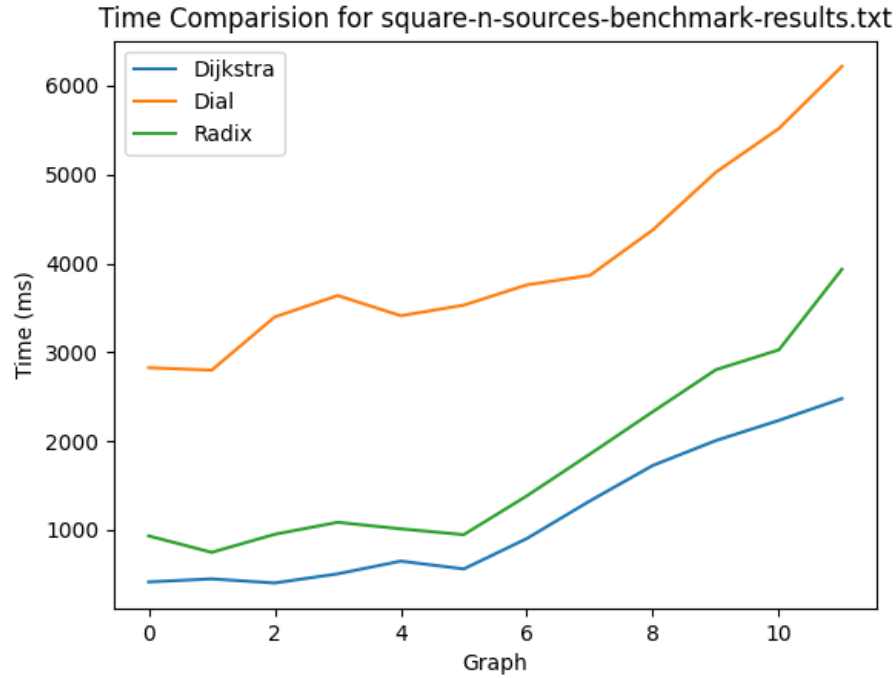
n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
1048576	4190208	1	974	97.4	4541	454.1	1030	103
1048576	4190208	4	781	78.1	1127	112.7	947	94.7
1048576	4190208	16	795	79.5	1052	105.2	1007	100.7
1048576	4190208	64	782	78.2	1026	102.6	965	96.5
1048576	4190208	256	776	77.6	1100	110	965	96.5
1048576	4190208	1024	775	77.5	1258	125.8	986	98.6
1048576	4190208	4096	775	77.5	1540	154	989	98.9
1048576	4190208	16384	749	74.9	1694	169.4	965	96.5
1048576	4190208	65536	762	76.2	1760	176	943	94.3
1048576	4190208	262144	755	75.5	1790	179	974	97.4
1048576	4190208	1048576	766	76.6	1783	178.3	973	97.3
1048576	4190208	4194302	744	74.4	1753	175.3	967	96.7
1048576	4190208	16777213	749	74.9	1756	175.6	2237	223.7
1048576	4190208	67108854	759	75.9	1784	178.4	2527	252.7
1048576	4190208	268435444	766	76.6	1804	180.4	2086	208.6
1048576	4190208	1073741250	764	76.4	1802	180.2	1526	152.6

Tabela 2: Porównanie średnich czasów obliczania odległości od źródła do wszystkich wierzchołków w grafie dla rodziny Square-C



n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
1024	3968	1023	416	0.0126953	2827	0.0862732	934	0.0285034
2025	7920	2025	450	0.0274658	2799	0.170837	749	0.0457153
4096	16128	4096	404	0.0493164	3398	0.414795	952	0.116211
8190	32398	8190	507	0.123779	3639	0.888428	1087	0.265381
16384	65024	16384	650	0.317383	3412	1.66602	1013	0.494629
32761	130320	32761	562	0.548828	3531	3.44824	948	0.925781
65536	261120	65536	904	1.76562	3759	7.3418	1384	2.70312
131044	522728	131044	1327	5.18359	3865	15.0977	1854	7.24219
262144	1046528	262144	1727	13.4922	4378	34.2031	2332	18.2188
524176	2093808	524176	2007	31.3594	5027	78.5469	2804	43.8125
1048576	4190208	1048576	2234	69.8125	5519	172.469	3028	94.625
2096704	8381024	2096704	2479	154.938	6218	388.625	3935	245.938

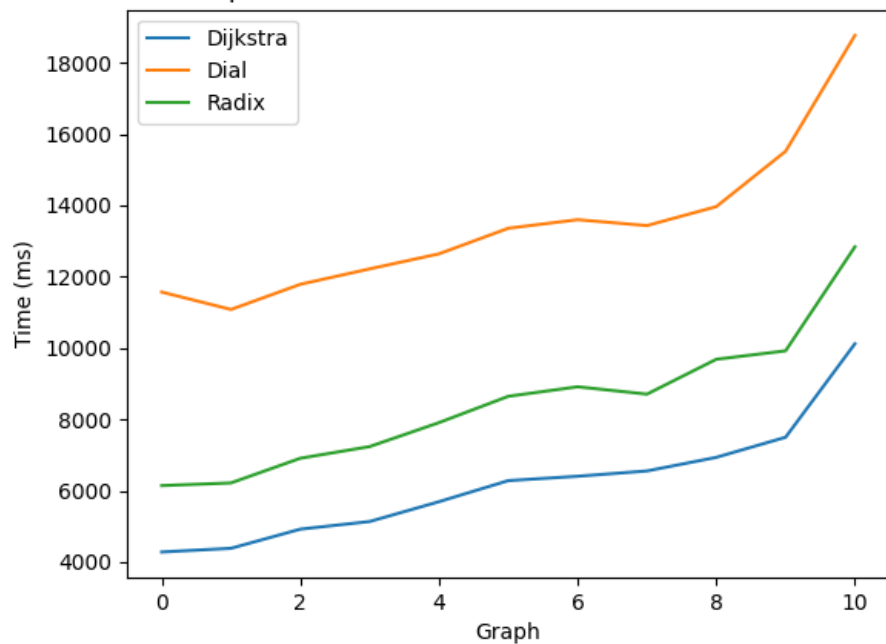
Tabela 3: Porównanie średnich czasów obliczania odległości od źródła do wszystkich wierzchołków w grafie dla rodziny Square-n



n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
264346	733846	36946	4282	8.04887	11570	21.7481	6143	11.547
321270	800172	94305	4385	10.0114	11082	25.3014	6218	14.1963
435666	1057066	137384	4922	15.2384	11787	36.4923	6910	21.3932
1070376	2712798	214013	5136	39.2061	12220	93.2824	7237	55.2443
1207945	2840208	128569	5692	49.069	12641	108.974	7909	68.181
1524453	3897636	63247	6283	68.2935	13359	145.207	8647	93.9891
1890815	4657742	215354	6405	86.5541	13599	183.77	8914	120.459
2758119	6885658	138911	6554	128.51	13434	263.412	8708	170.745
3598623	8778114	200760	6935	177.821	13965	358.077	9685	248.333
6262104	15248146	368855	7499	340.864	15518	705.364	9922	451
14081816	34292496	214013	10121	1012.1	18767	1876.7	12839	1283.9

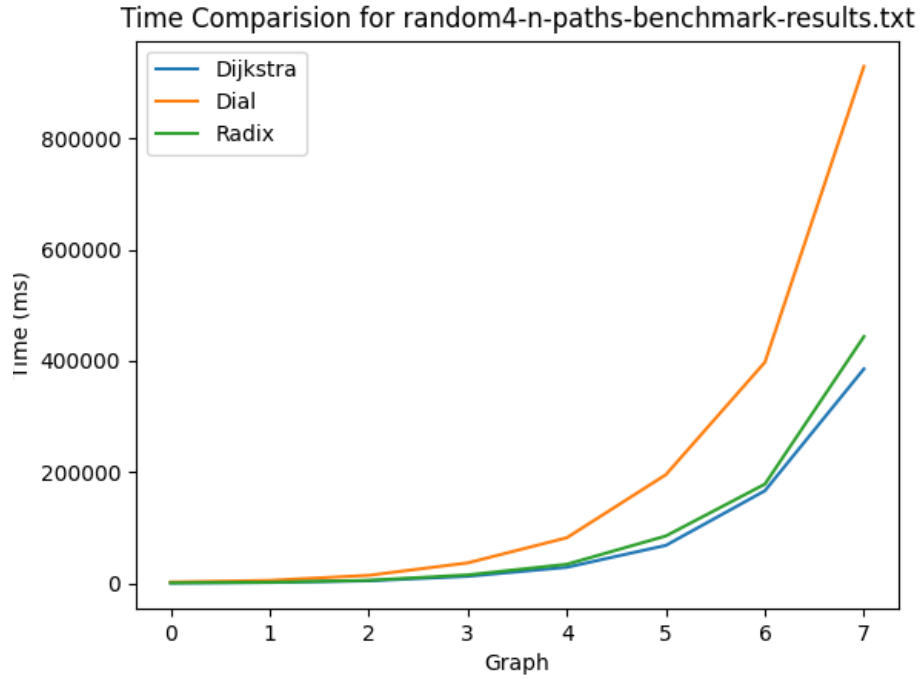
Tabela 4: Porównanie średnich czasów obliczania odległości od źródła do wszystkich wierzchołków w grafie dla rodziny USA-road-d

Time Comparision for usa-road-d-sources-benchmark-results.txt



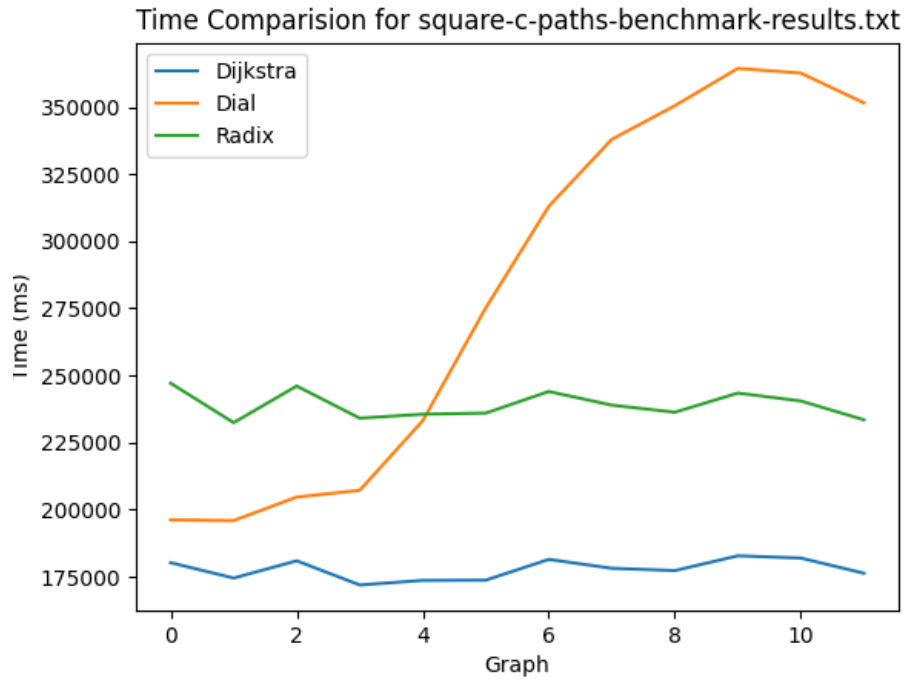
n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
16384	65536	16384	1086	1.086	2667	2.667	1293	1.293
32768	131072	32768	2103	2.103	5682	5.682	2347	2.347
65536	262144	65536	5045	5.045	14802	14.802	5862	5.862
131072	524288	131072	13066	13.066	37202	37.202	15575	15.575
262144	1048576	262143	29265	29.265	82364	82.364	34675	34.675
524288	2097152	524288	68648	68.648	195619	195.619	85519	85.519
1048576	4194304	1048576	166737	166.737	397589	397.589	178655	178.655
2097152	8388608	2097152	385685	385.685	929126	929.126	443761	443.761

Tabela 5: Porównanie średnich czasów obliczania ścieżek w grafie dla rodziny Random4-n



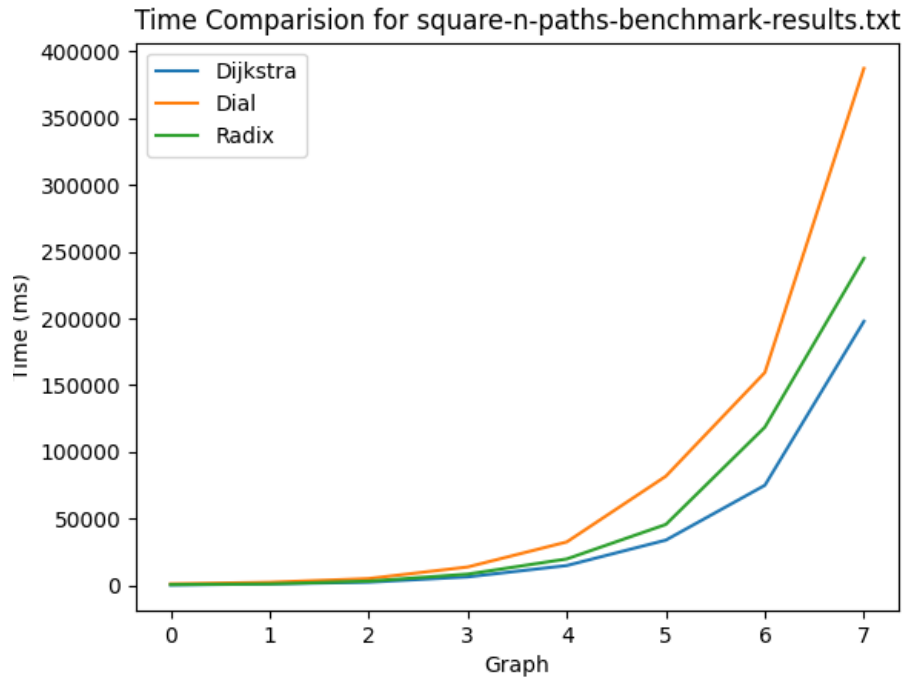
n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
1048576	4190208	1	180271	180.271	196199	196.199	247112	247.112
1048576	4190208	4	174535	174.535	195963	195.963	232423	232.423
1048576	4190208	16	180967	180.967	204700	204.7	246070	246.07
1048576	4190208	64	171959	171.959	207222	207.222	234092	234.092
1048576	4190208	256	173666	173.666	233154	233.154	235572	235.572
1048576	4190208	1024	173758	173.758	275192	275.192	235990	235.99
1048576	4190208	4096	181490	181.49	313012	313.012	244025	244.025
1048576	4190208	16384	178178	178.178	337969	337.969	238950	238.95
1048576	4190208	65536	177322	177.322	350491	350.491	236313	236.313
1048576	4190208	262144	182832	182.832	364390	364.39	243438	243.438
1048576	4190208	1048576	181981	181.981	362633	362.633	240491	240.491
1048576	4190208	4194302	176334	176.334	351553	351.553	233494	233.494

Tabela 6: Porównanie średnich czasów obliczania ścieżek w grafie dla rodziny Square-C



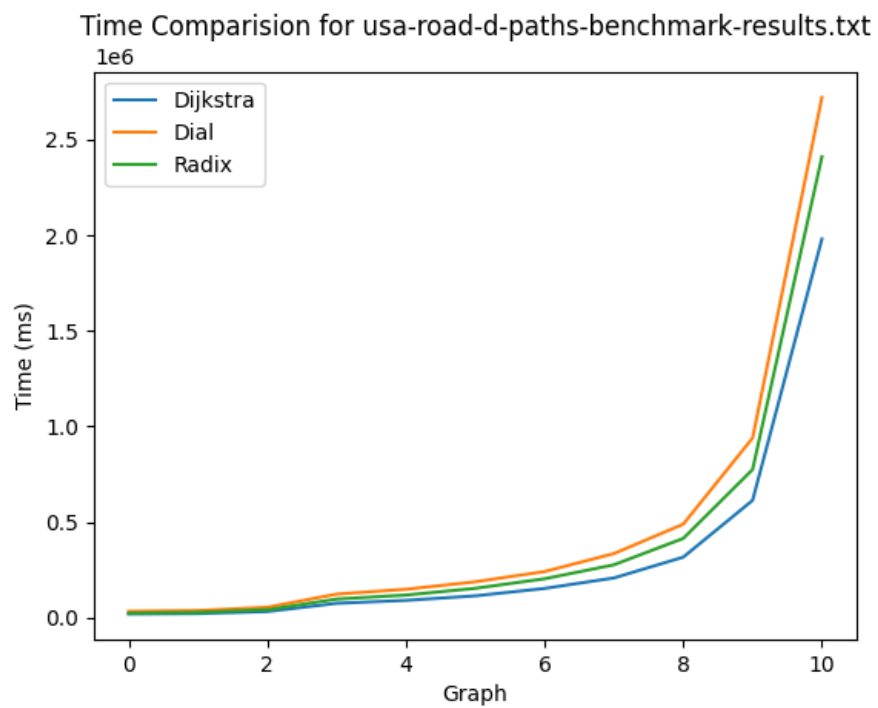
n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
16384	65024	16384	624	0.624	1463	1.463	937	0.937
32761	130320	32761	1466	1.466	2632	2.632	1432	1.432
65536	261120	65536	2722	2.722	5409	5.409	3453	3.453
131044	522728	131044	6682	6.682	14097	14.097	8758	8.758
262144	1046528	262144	15167	15.167	32819	32.819	20094	20.094
524176	2093808	524176	34175	34.175	81970	81.97	45881	45.881
1048576	4190208	1048576	75199	75.199	159549	159.549	118695	118.695
2096704	8381024	2096704	197801	197.801	386977	386.977	244922	244.922

Tabela 7: Porównanie średnich czasów obliczania ścieżek w grafie dla rodziny Square-n



n	m	c	Dijkstra (ms)	Avg (ms)	Dial (ms)	Avg (ms)	Radix (ms)	Avg (ms)
264346	733846	36946	18452	18.452	32721	32.721	22605	22.605
321270	800172	94305	20657	20.657	36320	36.32	26725	26.725
435666	1057066	137384	31163	31.163	53573	53.573	40675	40.675
1070376	2712798	214013	74291	74.291	122577	122.577	96444	96.444
1207945	2840208	128569	89586	89.586	147715	147.715	117468	117.468
1524453	3897636	63247	113337	113.337	186808	186.808	152627	152.627
1890815	4657742	215354	152154	152.154	240487	240.487	202611	202.611
2758119	6885658	138911	206878	206.878	334451	334.451	275755	275.755
3598623	8778114	200760	315844	315.844	488229	488.229	413751	413.751
6262104	15248146	368855	612877	612.877	939166	939.166	774259	774.259
14081816	34292496	214013	1980142	1980.14	2720603	2720.6	2410462	2410.46

Tabela 8: Porównanie średnich czasów obliczania ścieżek w grafie dla rodziny USA-road-d



5 Wnioski

Po czasie wykonywania algorytmów na grafach możemy zauważyć, że ważne jest dobranie odpowiedniego algorytmu do naszej struktury grafu.