Projektowanie Algorytmów i Metody Sztucznej Inteligencji
Projekt 1

20.03.2019r.

Nazwa kursu: Projektowanie algorytmów i metod sztucznej inteligencji
Prowadzący: dr inż. Łukasz Jeleń
Wykonał: Szymon Korczyński 24155
Termin: Środa 11-13

1. Wprowadzenie

Sortowanie to jeden z podstawowych problemów informatyki, który polega na uporządkowaniu zbioru danych w określonym porządku. Uporządkowane zbiory danych umożliwiają stosowanie bardziej wydajnych algorytmów np. wyszukiwanie. Również sortowanie jest wykorzystywane w celu prezentacji danych.

Algorytmy sortowania możemy skasyfikować według:

- złożoności obliczeniowej (ilości operacji potrzebnej do wykonania sortowania)
- złożoności pamięciowej
- stabilności (równe wartości nie zmieniają się kolejnością)

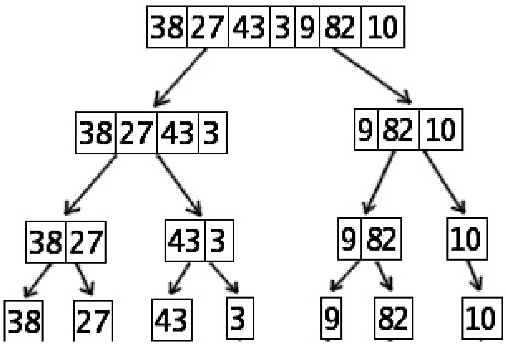
2. Badane algorytmy:
    a. Sortowanie przez scalanie

Algorytm sortowania przez scalanie polega na metodzie dziel i zwyciężaj. Oznacza to, że tablicę wejściową dzielimy na 2 tablice równej wielkości. Powtarzamy tę czynność do momentu uzyskania tablic 1-elementowych. Następnie porównujemy te tablice ze sobą, po czym tworzymy tablice 2-elementowe. Powtarzamy scalanie tablic(pamiętając, że uzyskane już tablice są posortowane) do momentu utrzymania tablicy o początkowej wielkości.

Złożoność obliczeniowa składa się z 2 części:

- Podziału tablicy do tablic 1-elementowych

Głębokość drzewa - ilość poziomów do uzyskania 1-elementowych tablic jest równa $\log_2 n$



- Scaleniu podzielonych tablic

Skoro podzielone tablice, są już uporządkowane to ilość operacji potrzebna do scalenia tych tablic wynosi n.

A więc, $O(n) = n \cdot \log(n)$

Złożoność obliczeniowa nie zależy od początkowego stanu tablicy wejściowej, ponieważ w każdym przypadku należy podzielić tablicę na 1-elementowe tablice.

    b. Sortowanie szybkie

Algorytm ten jest rozwinięciem wersji sortowania przez scalanie, dzięki wyliczeniu piwotu tablicy i podzieleniu jej względem zadanego piwotu (po prawej stronie znajdują się np. elementy większe od piwotu, natomiast po lewej od piwotu są tylko mniejsze od niego). Wprowadzenie piwotu zwiększa efektywność sortowania, ponieważ po podziale tablicy na 1-elementowe tablice, nie musimy ich scalać. Podzielone tablice są już posortowane w zadanej kolejności. Wyliczenie piwotu daje nam pewność o jego pozycji w tablicy wyjściowej.

Złożoność obliczeniowa składa się z 2 części:

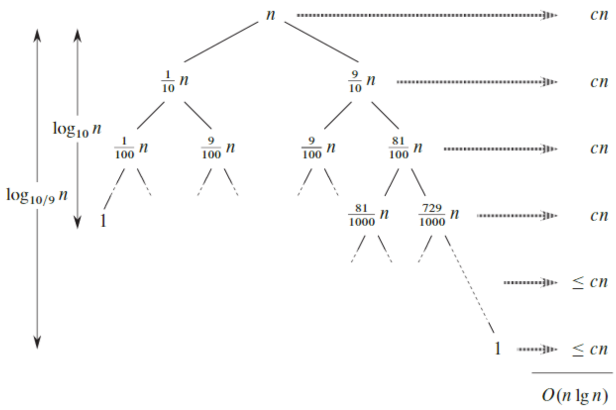
- Uszeregowaniu tablicy według piwota
- Podziale tych tablic według piwota

Najgorszy przypadek zachodzi, zawsze wybieramy piwot, który ma najwyższą wartość w dzielonej tablicy. Wówczas czas podziału wynosi O(n), a dzielone tablice mają wielkość n-1 oraz 0. Czyli
$T(n) = O(n) + T(n-1) + T(0) = O(n) + T(n-1 \quad = O(n) + O(n-1) + T(n-2) + \ldots = O(n) + O(n-1) + \ldots + O(2) + O(1) = O(n+1) + \cdots + O(n+1)$

$$T(n) = O(n+1) + \cdots + O(n+1)$$

Ilość takich elementów jest równa n/2 co daje złożoność $T(n) = \frac{n}{2} \cdot O(n+1)$.
Więc pesymistyczna złożoność równa jest $O(n) = n^2$
.

Przypadek średni:
Aby obliczyć czas działania średniego przypadku musimy założyć, że wybór piwotu za każdym razem jest niekorzystny np. 9 do 1. Wówczas $T(n) = T\left(\frac{9}{10}n\right) + T\left(\frac{1}{10}n\right) + O(n)$.
Po kilkukrotnym podziale tworzy się drzewo rekursji algorytmu.



Z przedstawionej grafiki wynika, że nawet po niekorzystnym podziale 9 do 1 czas działania algorytmu wynosi $O(n) = n \cdot \log(n)$

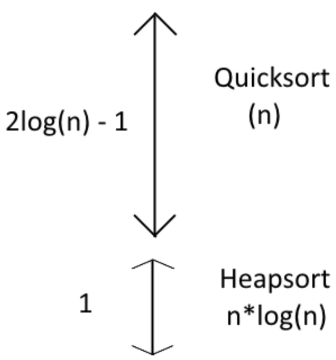
c. Sortowanie introspektywne:

Sortowanie introspektywne jest sortowaniem hybrydowym tzn. wykorzystuje dwa algorytmy sortowania: sortowanie przez kopcowanie oraz sortowanie szybkie.

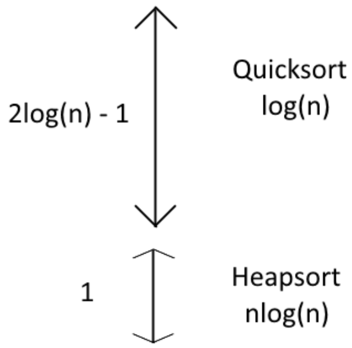
Złożoność obliczeniowa składa się z 2 części:

- sortowania dużych tablic przy użyciu sortowania szybkiego
- sortowania małych tablic przy użyciu sortowania przez kopcowanie

Najgorszy przypadek:



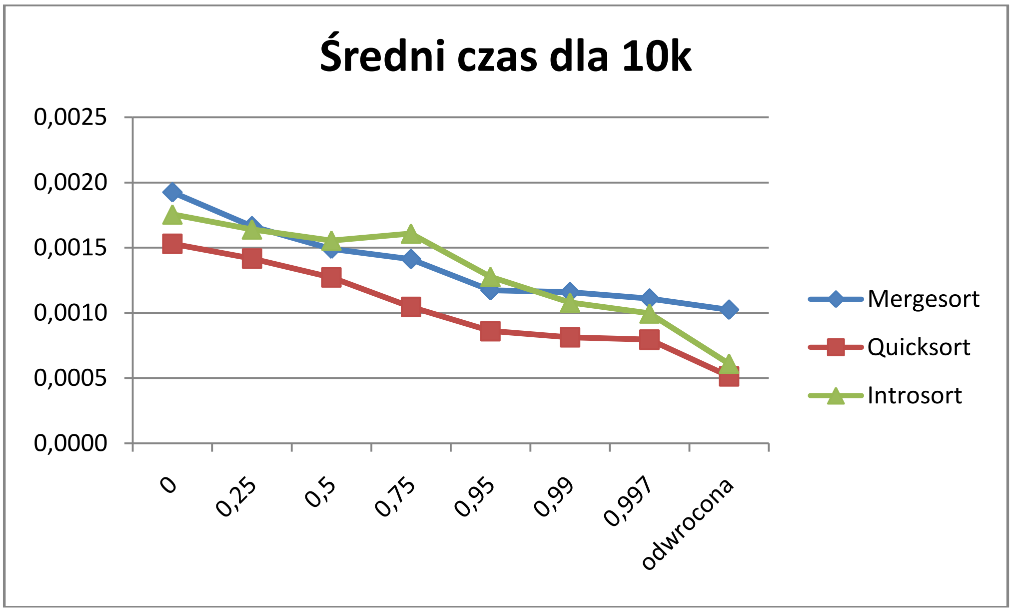
Na przedstawionej grafice widać, że $T(n) = (2\log(n) - 1) \cdot n + n \cdot \log(n)$,
czyli $O(n) = n \cdot \log(n)$
W przypadku średnim:

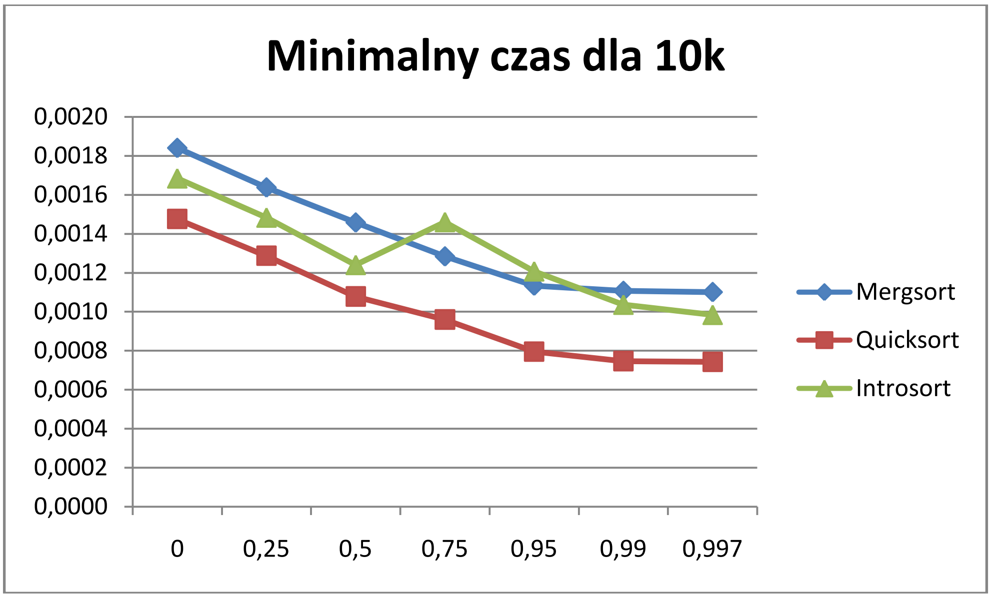


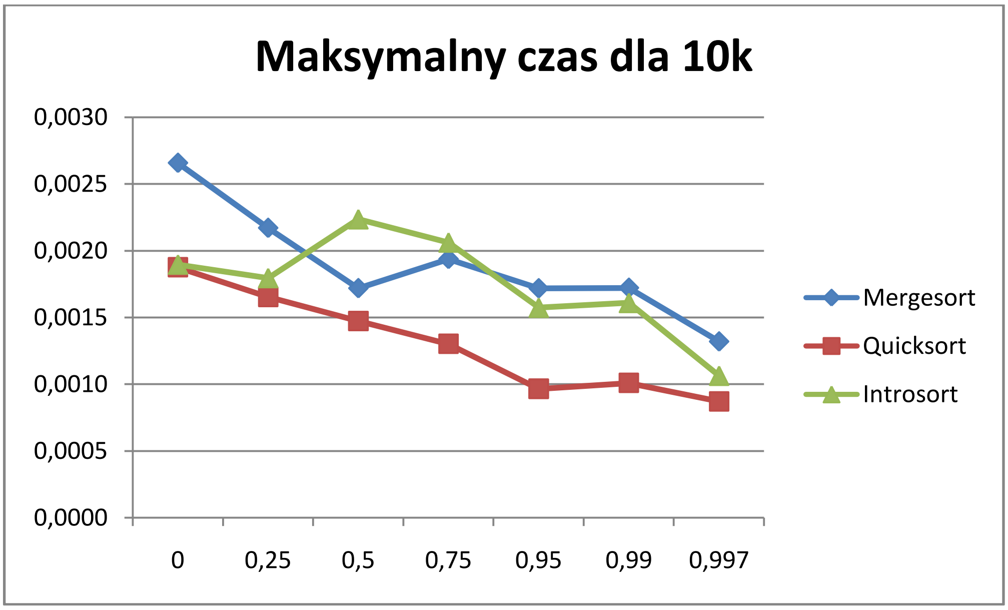
Na przedstawionej grafice widać, że $T(n) = (2\log(n) - 1) \cdot \log(n) + n \cdot \log(n) = 2\log(n)^2 - \log(n) + n \cdot \log(n)$,
czyli $O(n) = n \cdot \log(n)$

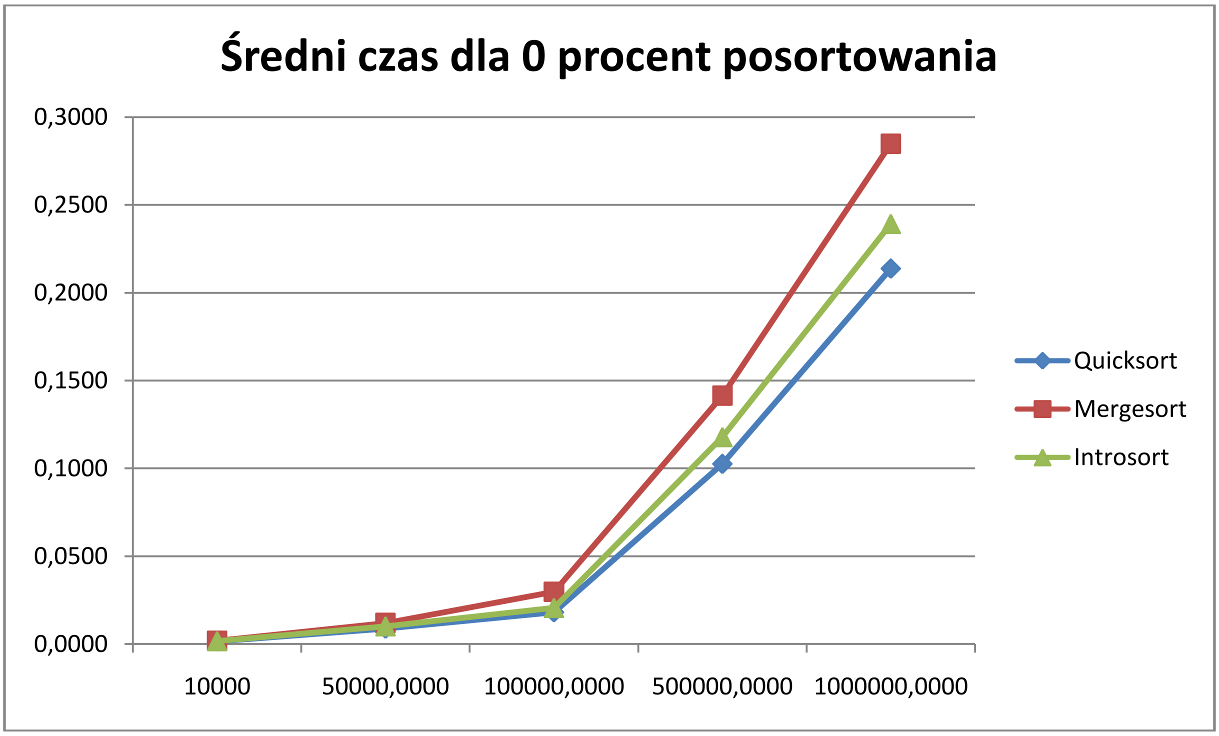
3. Wyniki

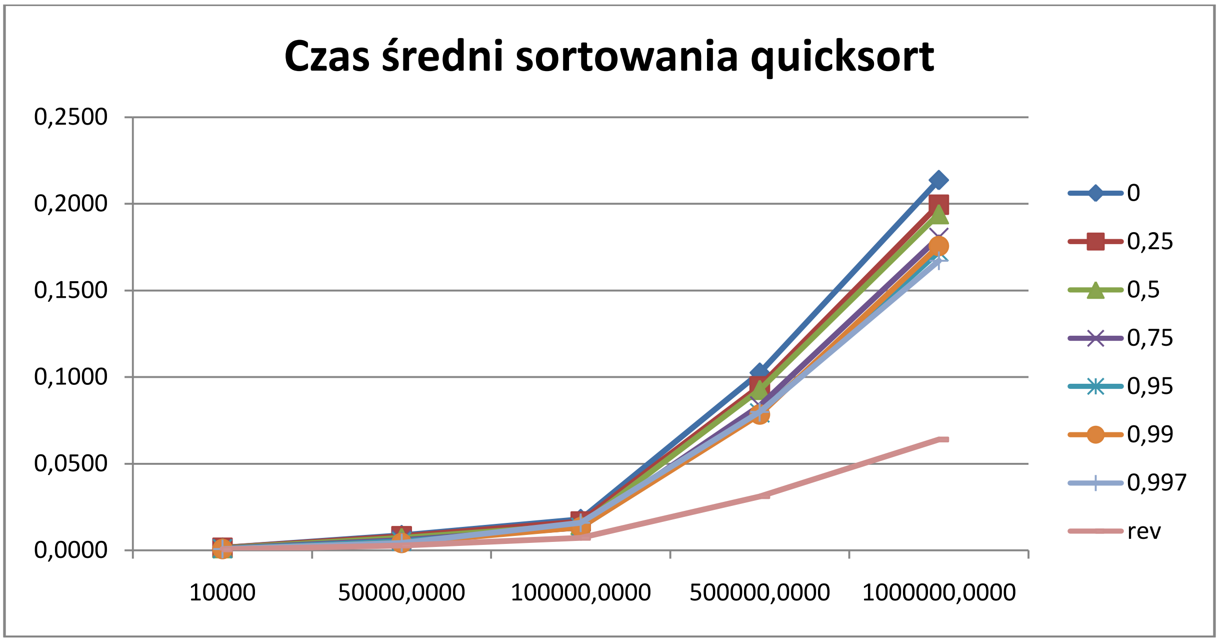
Poziom posortowania tablicy	Mergesort			Quicksort			Introsort		
	min	avg	max	min	avg	max	min	avg	max
	Dla 10 tyś. elementów								
0	0,0018	0,0019	0,0027	0,0015	0,0015	0,0019	0,0017	0,0018	0,0019
0,25	0,0016	0,0017	0,0022	0,0013	0,0014	0,0017	0,0015	0,0016	0,0018
0,5	0,0015	0,0015	0,0017	0,0011	0,0013	0,0015	0,0012	0,0016	0,0022
0,75	0,0013	0,0014	0,0019	0,0010	0,0010	0,0013	0,0015	0,0016	0,0021
0,95	0,0011	0,0012	0,0017	0,0008	0,0009	0,0010	0,0012	0,0013	0,0016
0,99	0,0011	0,0012	0,0017	0,0007	0,0008	0,0010	0,0010	0,0011	0,0016
0,997	0,0011	0,0011	0,0013	0,0007	0,0008	0,0009	0,0010	0,0010	0,0011
odwrocona		0,0010			0,0005			0,0006	
	Dla 50 tyś. elementów								
0	0,0107	0,0120	0,0149	0,0085	0,0087	0,0093	0,0096	0,0102	0,0132
0,25	0,0095	0,0098	0,0128	0,0073	0,0080	0,0090	0,0085	0,0096	0,0125
0,5	0,0085	0,0087	0,0110	0,0063	0,0072	0,0092	0,0074	0,0090	0,0129
0,75	0,0074	0,0079	0,0114	0,0054	0,0059	0,0068	0,0087	0,0096	0,0129
0,95	0,0065	0,0077	0,0113	0,0045	0,0051	0,0059	0,0073	0,0077	0,0094
0,99	0,0067	0,0095	0,0131	0,0041	0,0042	0,0053	0,0061	0,0067	0,0092
0,997	0,0068	0,0095	0,0122	0,0041	0,0043	0,0048	0,0058	0,0062	0,0078
odwrocona		0,0083			0,0027			0,0033	
	Dla 100 tyś. elementów								
0	0,0232	0,0298	0,0364	0,0176	0,0181	0,0201	0,0200	0,0207	0,0236
0,25	0,0208	0,0278	0,0326	0,0153	0,0166	0,0200	0,0176	0,0190	0,0230
0,5	0,0180	0,0251	0,0306	0,0130	0,0147	0,0174	0,0156	0,0187	0,0253
0,75	0,0156	0,0208	0,0274	0,0124	0,0131	0,0154	0,0219	0,0228	0,0274
0,95	0,0140	0,0159	0,0246	0,0118	0,0132	0,0181	0,0210	0,0218	0,0261
0,99	0,0137	0,0154	0,0249	0,0122	0,0133	0,0161	0,0214	0,0231	0,0329
0,997	0,0137	0,0150	0,0248	0,0128	0,0159	0,0352	0,0219	0,0248	0,0340
odwrocona		0,0125			0,0072			0,0081	
	Dla 500 tyś. elementów								
0	0,1279	0,1414	0,1977	0,0971	0,1025	0,1227	0,1108	0,1179	0,1311
0,25	0,1155	0,1318	0,1807	0,0902	0,0947	0,1075	0,1035	0,1111	0,1329
0,5	0,1056	0,1177	0,1620	0,0856	0,0926	0,1114	0,1006	0,1071	0,1246
0,75	0,0954	0,1122	0,1606	0,0803	0,0833	0,0897	0,0931	0,0993	0,1150
0,95	0,0890	0,0938	0,1218	0,0769	0,0793	0,0845	0,0888	0,0933	0,0992
0,99	0,0881	0,0930	0,1260	0,0758	0,0784	0,0867	0,0887	0,0946	0,1241
0,997	0,0882	0,0929	0,1196	0,0759	0,0798	0,0945	0,0868	0,0926	0,1025
odwrocona		0,0726			0,0311			0,0363	
	Dla 1 miliona elementów								
0	0,2684	0,2848	0,3987	0,2034	0,2137	0,2547	0,2307	0,2392	0,2581
0,25	0,2450	0,2695	0,3649	0,1909	0,1995	0,2328	0,2200	0,2273	0,2437
0,5	0,2288	0,2568	0,3150	0,1791	0,1941	0,2266	0,2113	0,2207	0,2392
0,75	0,2090	0,2279	0,3367	0,1693	0,1806	0,2227	0,1953	0,2084	0,2338
0,95	0,1961	0,2112	0,2729	0,1636	0,1723	0,1987	0,1879	0,1985	0,2294
0,99	0,1898	0,2009	0,2297	0,1623	0,1756	0,2134	0,1889	0,2042	0,2471
0,997	0,1897	0,2042	0,2757	0,1609	0,1671	0,1996	0,1907	0,2122	0,2685
odwrocona		0,1469			0,0640			0,0750	

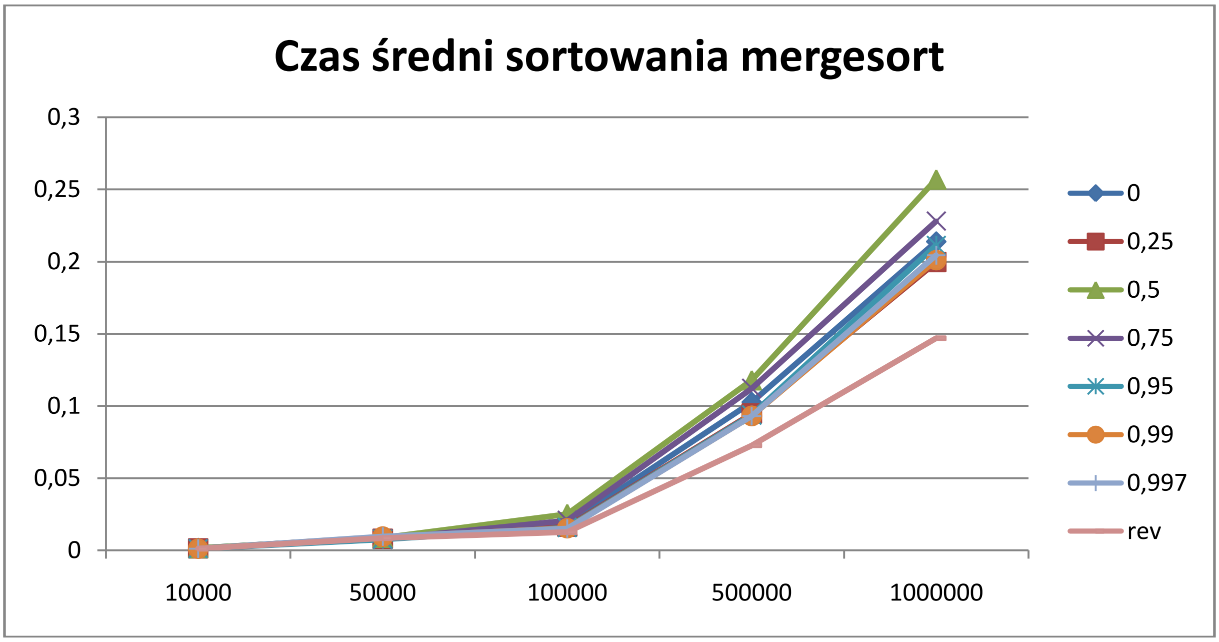

Średni czas dla 10k
0,0025
0,0020
0,0015
0,0010
0,0005
0,0000
0
0,25
0,5
0,75
0,95
0,99
0,997
odwrocona
Mergesort
Quicksort
Introsort

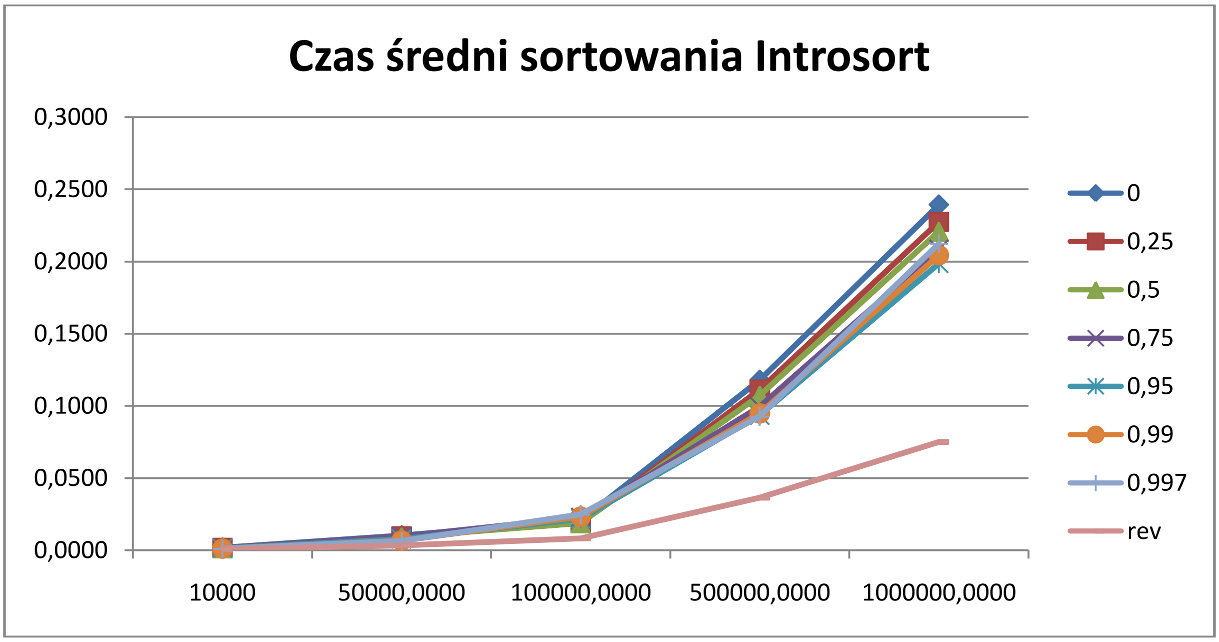

Minimalny czas dla 10k
0,0020
0,0018
0,0016
0,0014
0,0012
0,0010
0,0008
0,0006
0,0004
0,0002
0,0000
0
0,25
0,5
0,75
0,95
0,99
0,997
Mergsort
Quicksort
Introsort


Maksymalny czas dla 10k
0,0030
0,0025
0,0020
0,0015
0,0010
0,0005
0,0000
0
0,25
0,5
0,75
0,95
0,99
0,997
Mergesort
Quicksort
Introsort


Średni czas dla 0 procent posortowania
0,3000
0,2500
0,2000
0,1500
0,1000
0,0500
0,0000
10000
50000,0000
100000,0000
500000,0000
1000000,0000
Quicksort
Mergesort
Introsort


Czas średni sortowania quicksort
0,2500
0,2000
0,1500
0,1000
0,0500
0,0000
10000
50000,0000
100000,0000
500000,0000
1000000,0000
0
0,25
0,5
0,75
0,95
0,99
0,997
rev





4. Wnioski

Dłuższy czas działania algorytmu Introsort może być spowodowany przez potrzebę kopiowania elementów do tablicy pomocniczej w celu posortowania algorytmem Heapsort.

Czas średni dla sortowania Quicksort i Introsort jest dużo mniejszy od czasu średniego sortowania w pozostałych parametrach początkowych tablicy, wynika to z wyboru na każdym etapie najlepszego piwotu.

5. Literatura

1. Cormen T., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów
2. https://en.wikipedia.org/wiki/Introsort
3. https://en.wikipedia.org/wiki/Quicksort
4. https://en.wikipedia.org/wiki/Merge_sort