

Sprawozdanie PAMSI

Algorytmy sortowania quicksort i mergesort na liście jednokierunkowej dla
różnego wyboru *pivota* i różnej ilości elementów.

- **Wstęp**

Algorytmy mergesort i quicksort należą do grupy tak zwanych algorytmów szybkich. Oba algorytmy stosują metodę „dziel i zwyciężaj” polegającą na rekurencyjnym dzieleniu problemu na pomniejsze problemy aż do momentu maksymalnego uproszczenia struktury.

Złożoność obliczeniowa algorytmu quicksort zawiera się pomiędzy $O(n \log n)$ a $O(n^2)$ i w dużej mierze zależy od wyboru *pivota* (elementu dzielącego zbiór na dwa podzbiory). Z tego powodu algorytm quicksort klasyfikowany jest jako algorytm niestabilny.

Złożoność obliczeniowa algorytmu mergesort wynosi $O(n \log n)$.

Różnicą pomiędzy algorytmem quicksort i mergesort, jest fakt że mergesort do działania potrzebuje dodatkowej tablicy do której przepisuje tymczasowo wczytane dane. Wymaga to więc alokowania dodatkowej pamięci rzędu $O(n)$.

- Tabele pomiarowe

Liczba Elementów	Pivot w środku		Pivot jako pierwszy element	
	Quicksort	Mergesort	Quicksort	Mergesort
	Czas Sortowania [s]	Czas Sortowania [s]	Czas Sortowania [s]	Czas Sortowania [s]
10	0,00003	0,00010	0,00034	0,00014
	0,00002	0,00010	0,00035	0,00010
	0,00003	0,00013	0,00029	0,00013
	0,00004	0,00012	0,00032	0,00012
	0,00003	0,00010	0,00031	0,00010
	0,00003	0,00012	0,00091	0,00013
	0,00002	0,00010	0,00038	0,00011
	0,00003	0,00013	0,00028	0,00013
	0,00003	0,00011	0,00041	0,00011
	0,00003	0,00013	0,00033	0,00013
100	0,001435	0,000885	0,00131	0,00082
	0,001014	0,000885	0,00137	0,00086
	0,001234	0,000868	0,00137	0,00090
	0,000772	0,000864	0,00131	0,00086
	0,001236	0,000517	0,00099	0,00089
	0,001236	0,000529	0,00131	0,00086
	0,000772	0,000874	0,00131	0,00090
	0,000742	0,000882	0,00128	0,00087
	0,001234	0,000970	0,00099	0,00087
	0,000926	0,000895	0,00131	0,00088
1000	0,096850	0,060571	0,10051	0,06284
	0,100549	0,059811	0,10035	0,06117
	0,100929	0,060576	0,99018	0,06018
	0,099115	0,062311	0,10171	0,05981
	0,096588	0,060277	0,09918	0,06058
	0,096534	0,061167	0,09913	0,06058
	0,096887	0,060181	0,09948	0,06231
	0,098263	0,060942	0,10086	0,05727
	0,097320	0,060256	0,09991	0,06117
	0,096113	0,057265	0,09871	0,06028
5000	3,85666	2,13242	3,93273	2,16425
	3,12327	2,15279	3,95124	2,14872
	3,84700	2,14872	3,93587	2,11110
	2,91639	2,11110	4,01977	2,12943
	3,88715	2,12943	3,94992	2,12943
	3,55261	2,13267	3,92358	2,13267
	3,84409	2,12980	3,92653	2,12723
	3,84342	2,12723	3,94377	2,11088
	3,85843	2,11088	3,92693	2,14872
	3,88132	2,12005	3,96786	2,13267
10000	17,6335	9,39836	18,34914	9,52192
	16,7998	9,41348	17,51522	9,53704
	17,0283	9,35166	17,74394	9,47522
	17,7409	9,38618	18,45651	9,50974
	16,7174	9,36215	17,43307	9,48571
	16,7547	9,36348	17,47037	9,48704
	17,1928	9,37319	17,90844	9,49675
	17,7889	9,38461	18,50454	9,50817
	16,5626	9,38115	17,27829	9,50471
	16,9016	9,38275	17,61726	9,50631

Liczba Elementów	Pivot w środku		Pivot jako pierwszy element	
	Quicksort	Mergesort	Quicksort	Mergesort
	Czas Sortowania [s]	Czas Sortowania [s]	Czas Sortowania [s]	Czas Sortowania [s]
20000	72,485	40,668	74,757	39,614
	71,421	40,586	73,692	39,533
	72,236	40,682	74,507	39,629
	72,252	40,599	74,523	39,546
	71,982	40,703	74,254	39,650
	72,135	40,329	74,406	39,276
	72,425	40,891	74,697	39,838
	71,133	40,759	73,404	39,706
	71,935	40,623	74,206	39,569
	72,002	40,782	74,273	39,728
30000	162,134	96,845	172,627	98,771
	162,192	96,134	172,426	97,923
	163,151	96,843	173,386	98,770
	161,345	96,342	171,579	98,268
	162,986	96,345	173,221	98,271
	163,102	96,614	173,336	98,540
	162,530	96,426	172,765	98,352
	161,986	96,514	172,221	98,440
	162,258	96,246	172,492	98,173
	162,937	96,213	173,171	98,139
40000	310,135	180,135	324,964	181,369
	310,192	180,193	325,434	181,427
	311,153	181,153	324,654	182,388
	309,346	179,346	325,615	180,581
	310,988	180,988	323,808	182,223
	311,102	181,103	325,450	182,337
	310,532	180,532	325,564	181,767
	309,987	179,988	324,994	181,222
	310,258	180,259	324,450	181,493
	310,938	180,938	324,720	182,173
50000	504,928	295,077	518,243	294,410
	504,856	295,005	518,171	294,338
	504,857	295,006	518,172	294,339
	504,157	294,306	517,472	293,639
	504,191	294,340	517,506	293,673
	504,824	294,973	518,139	294,306
	504,207	294,356	517,522	293,689
	504,810	294,959	518,125	294,292
	504,911	295,060	518,226	294,393
	504,355	294,504	517,670	293,837
100000	1607,93	1285,37	2132,37	1287,88
	1607,38	1301,24	2136,12	1303,75
	1611,02	1300,51	2146,41	1303,02

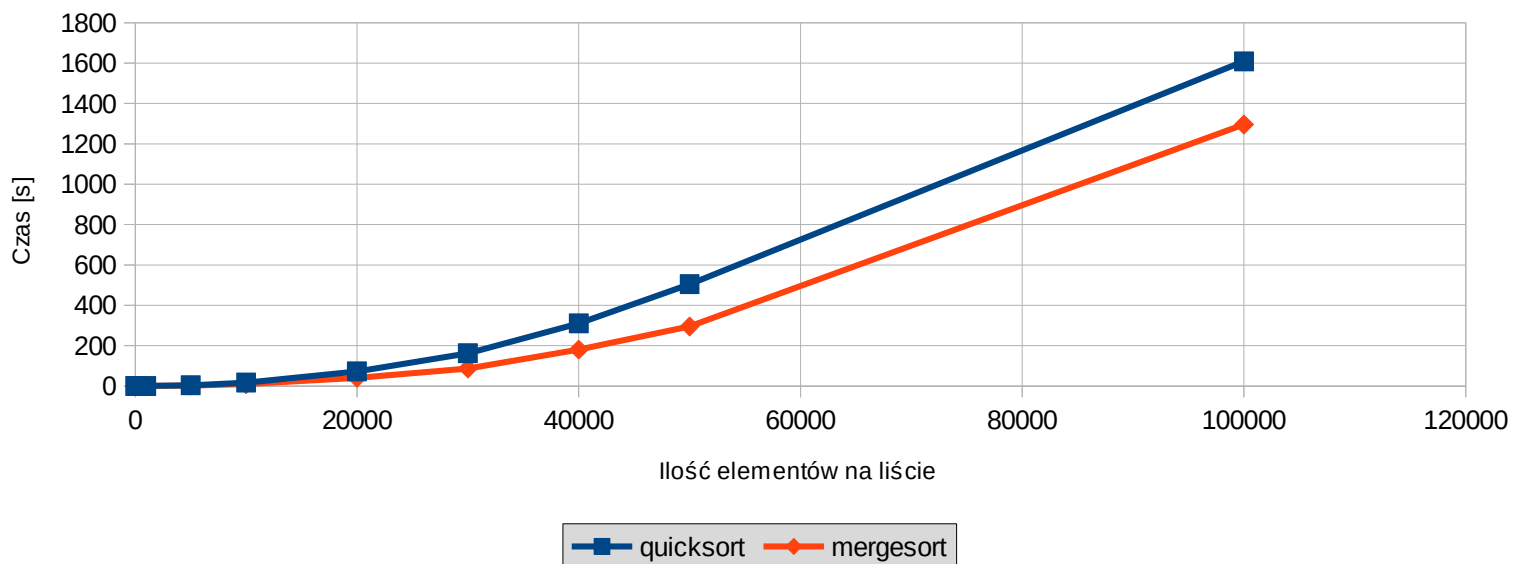
Tabela 1

Ilość Elementów	Pivot w środku		Pivot jako pierwszy element	
	Quicksort	Mergesort	Quicksort	Mergesort
	Średni Czas [s]	Średni Czas [s]	Średni Czas [s]	Średni Czas [s]
10	0,00003	0,00011	0,00039	0,00012
100	0,00106	0,00082	0,00126	0,00087
1000	0,09791	0,06034	0,18900	0,06062
5000	3,66103	2,12951	3,94782	2,13351
10000	17,11205	9,37970	17,82768	9,50326
20000	72,00054	40,66232	74,27184	39,60865
30000	162,46207	86,83886	172,72243	98,36470
40000	310,46302	180,46342	324,96526	181,69792
50000	504,60960	294,75860	517,92410	294,09140
100000	1608,77367	1295,70667	2138,30000	1298,22027

Tabela 2

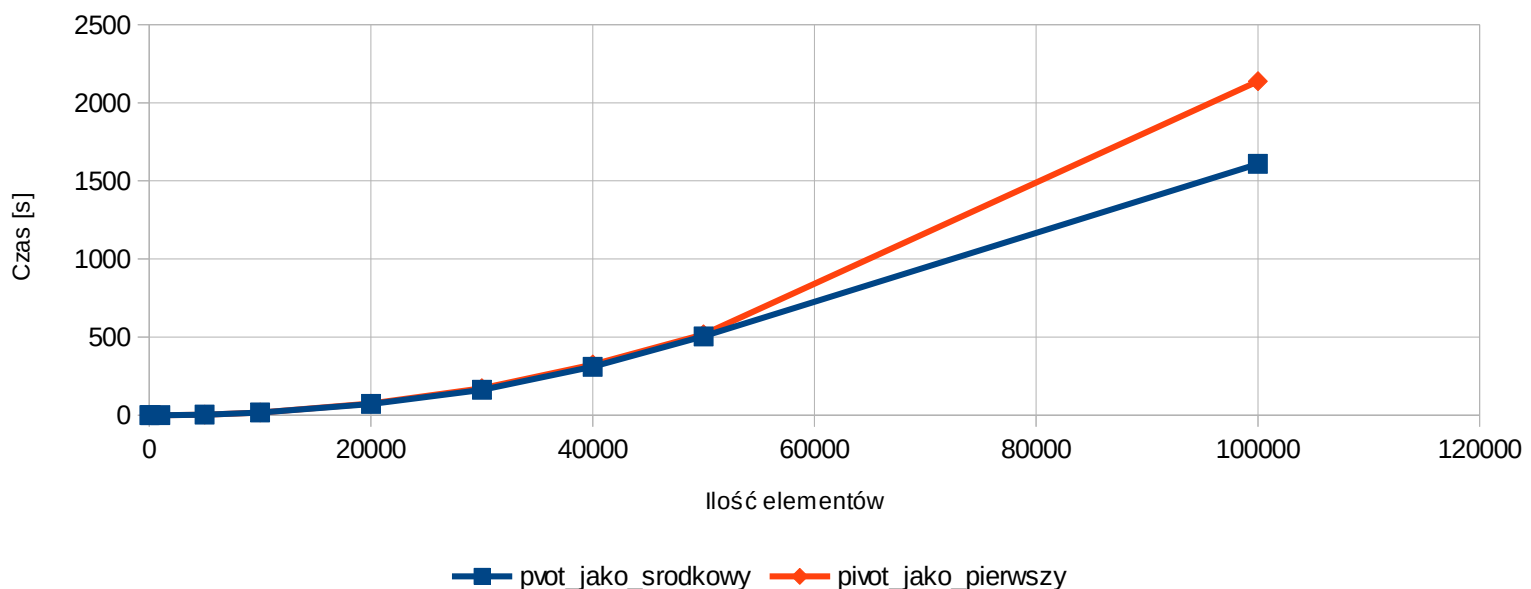
Wykres I

Porównanie algorytmów Quicksort i Mergesort dla pivotu jako środkowy element



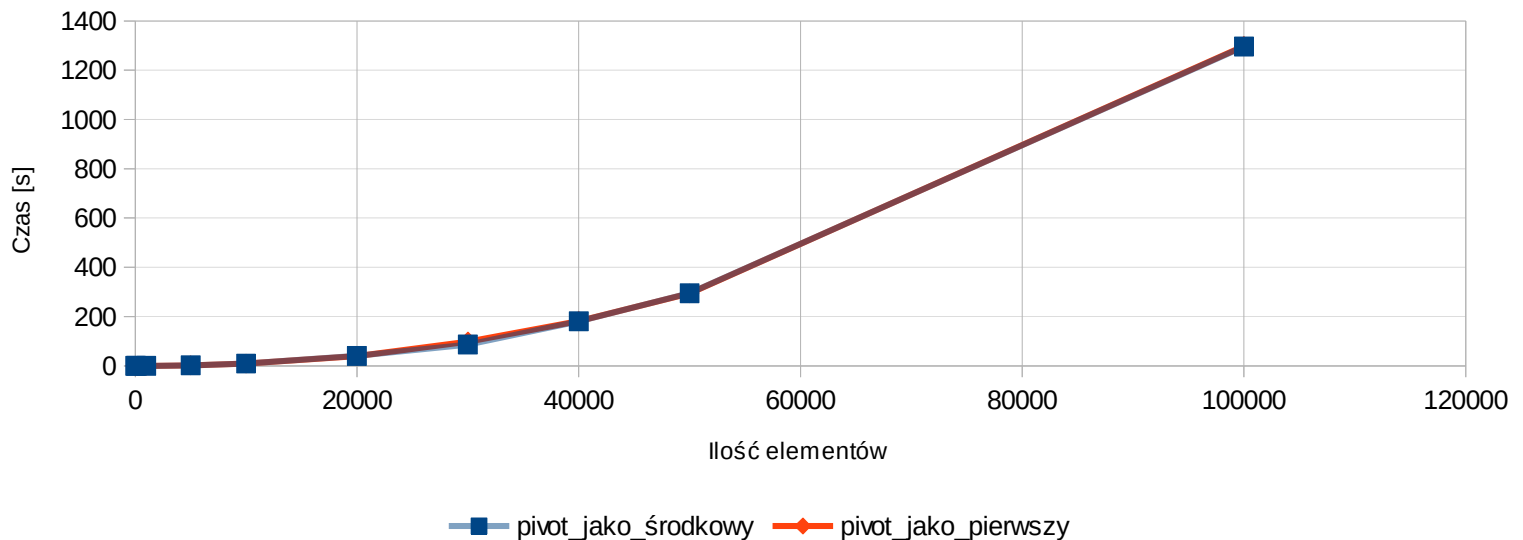
Wykres II

Porównanie czasu wykonywania algorytmów Quicksort w zależności od wyboru pivotu



Wykres III

Porównanie czasu wykonywania algorytmów Mergesort w zależności od wyboru pivotu



- **Wnioski**

Teoretyczna złożoność obliczeniowa algorytmu quicksort waha się pomiędzy przypadkiem optymistycznym $O(n) = n \log(n)$ a pesymistycznym $O(n) = n^2/2$. Na tak dużą rozbieżność pomiędzy stopniem złożoności obliczeniowej wpływa wybór pivotu, który decyduje o sposobie podziału sortowanego zbioru. Odpowiedni pivot może więc sprawić, że zbiór podzielimy na dwie równe części (co jest przypadkiem skrajnie optymistycznym), lub na zbiór jednoelementowy i zbiór zawierający resztę elementów (skrajnie pesymistyczny).

Na podstawie przeprowadzonych pomiarów w *Tabela 2* i *Wykres 1*, widzimy że algorytm quicksort wykonuje swoje zadanie dłużej niż algorytm mergesort. Spodziewanym rezultatem były bardzo zbliżone do siebie wyniki. Taką rozbieżność można wytłumaczyć dwoma faktami.

1. Implementacja algorytmu quicksort może być „nieidealna”.
2. Sortowanymi obiektami są listy jednokierunkowe. Za każdym razem jeżeli chcemy dostać informację na temat jakiegokolwiek elementu listy, musimy najpierw dostać się do tego elementu po wskaźnikach.

Można jednak zauważyć, że złożoność obu algorytmów jest większa niż $O(n \log n)$, ale też mniejsza niż $O(n^2)$.

Na podstawie *Wykres II* widać, że zgodnie z przewidywaniami zmiana wyboru pivotu niekorzystnie wpłynęła na czas wykonywania algorytmu quicksort. Patrząc na *Wykres II* i *Tabela 2* widać, że złożoność obliczeniowa algorytmu gdy pivot jest pierwszym elementem jest zbliżona do $O(n^2)$. Na podstawie *Wykres III* i *Tabela 2* można wywnioskować, że mimo zmiany położenia pivotu dla algorytmu mergesort otrzymujemy bardzo zbliżone pomiary. Fakt ten potwierdza to stabilność algorytmu.