

## Sortowanie

Tabela 1 Czasy wykonania algorytmu quicksort podane w sekundach

Ilość elementów do posortowania:				
10	$10^2$	$10^3$	$10^6$	$10^8$
0,000002	0,00001	0,000131	0,1596	16,5939
0,000005	0,000041	0,000419	0,150416	16,7757
0,000005	0,000033	0,000394	0,148797	16,7919
0,000002	0,000013	0,000179	0,147321	16,4486
0,000006	0,000034	0,000416	0,14582	25,3275
0,000002	0,000011	0,000133	0,149245	16,3763
0,000006	0,000038	0,000399	0,152103	16,7014
0,000005	0,000036	0,000394	0,151145	16,403
0,000004	0,000031	0,00039	0,151635	18,2667
0,000006	0,000066	0,000395	0,230397	25,9497

Tabela 2 Średnie czasy wykonania algorytmu quicksort podane w sekundach

Ilość elementów do posortowania:				
10	$10^2$	$10^3$	$10^6$	$10^8$
0,0000043	0,0000313	0,000325	0,1586479	18,56347

Wykres 1 Zależność czasu od liczby elementów do posortowania dla algorytmu quicksort

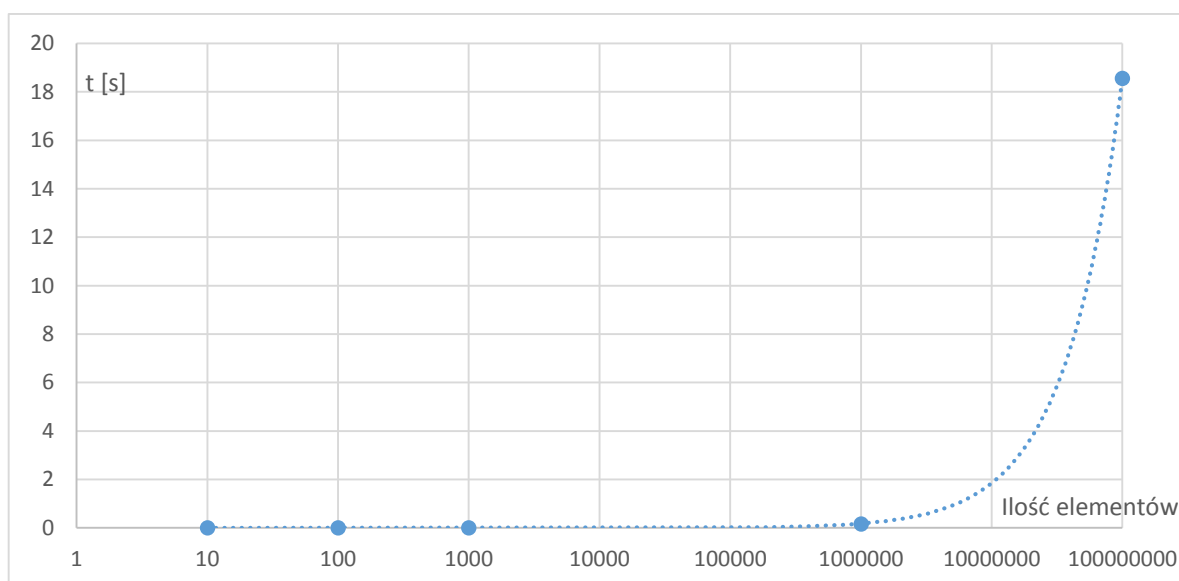


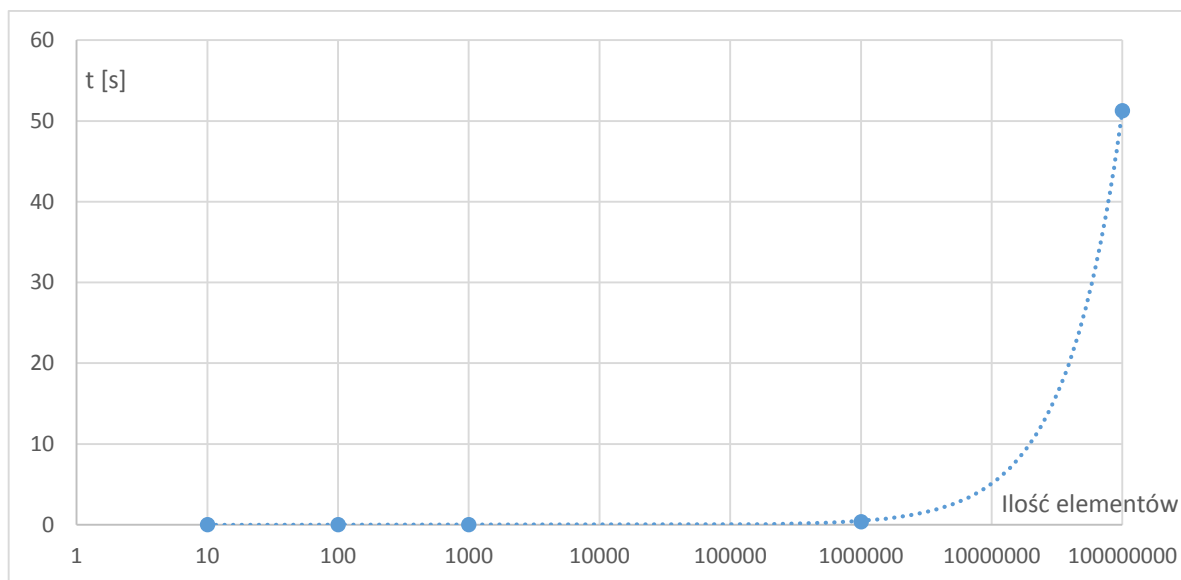
Tabela 3 Czasy wykonania algorytmu merge podane w sekundach

Lista elementów do posortowania:				
10	$10^2$	$10^3$	$10^6$	$10^8$
0,000012	0,000069	0,00072	0,321244	38,7171
0,000012	0,000068	0,000725	0,338109	58,0546
0,000013	0,00007	0,000738	0,262644	36,9222
0,000005	0,000022	0,00023	0,328964	42,7957
0,000007	0,000037	0,00036	0,508155	57,9807
0,000012	0,000069	0,00073	0,328823	58,9523
0,000013	0,000069	0,000805	0,3233	37,3202
0,000006	0,000035	0,000364	0,325175	67,5647
0,000013	0,000068	0,000725	0,321562	37,616
0,000012	0,000068	0,000726	0,325763	76,7701

Tabela 4 Średnie czasy wykonania algorytmu merge podane w sekundach

Ilość elementów do posortowania:				
10	$10^2$	$10^3$	$10^6$	$10^8$
0,0000105	0,0000575	0,0006123	0,3383739	51,26936

Wykres 2 Zależność czasu od liczby elementów do posortowania dla algorytmu merge



## Wnioski:

Złożoność czasową algorytmu quicksort można podzielić na trzy przypadki:

1. Optymistyczny – ma miejsce, gdy pivot jest medianą sortowanego fragmentu zbioru. Złożoność czasowa w tym przypadku wynosi  $n \log_2 n$ .
2. Średni – zachodzi, gdy pivot jest dowolną wartością z równomiernego rozkładu prawdopodobieństwa. Złożoność czasowa wynosi w przybliżeniu  $1,39n \log_2 n$ .
3. Pesymistyczny – jest wtedy, gdy pivot jest ciągle najmniejszą lub największą wartością z tablicy, przez co algorytm ma złożoność czasową wynoszącą  $\frac{n^2}{2}$ .

Zaimplementowany program przypisywał tablicy wartości z zakresu 0-1000 za pomocą funkcji rand() oraz jako pivot wybierał pierwszy element tablicy. Wyniki pomiarów najbliższe są złożoności czasowej z przypadku średniego.

Wybór pivota ma wpływ na złożoność czasową algorytmu. Wybierając jeden ze skrajnych elementów narażonym się jest na pesymistyczną złożoność algorytmu w przypadku, gdy elementy są już posortowane, ale np. na odwrót. Rozwiązaniem może być funkcja obliczająca medianę z zadanej ilości elementów zbioru, dzięki czemu można wykluczyć na trafienie pesymistycznego przypadku oraz zwiększyć prawdopodobieństwo na optymistyczny. Taki sposób zwiększa jednak złożoność czasową algorytmu, dlatego powinien być stosowany początkowych podziałach, gdy wybór pivota ma największy wpływ na czas wykonania algorytmu.

Algorytm sortowania merge okazał się być wolniejszy niż quicksort, przy czym wraz ze zwiększaniem ilości elementów różnica między nimi zwiększała się.