Sprawozdanie z zajęć laboratoryjnych – sortowanie

Sortowanie szybkie:

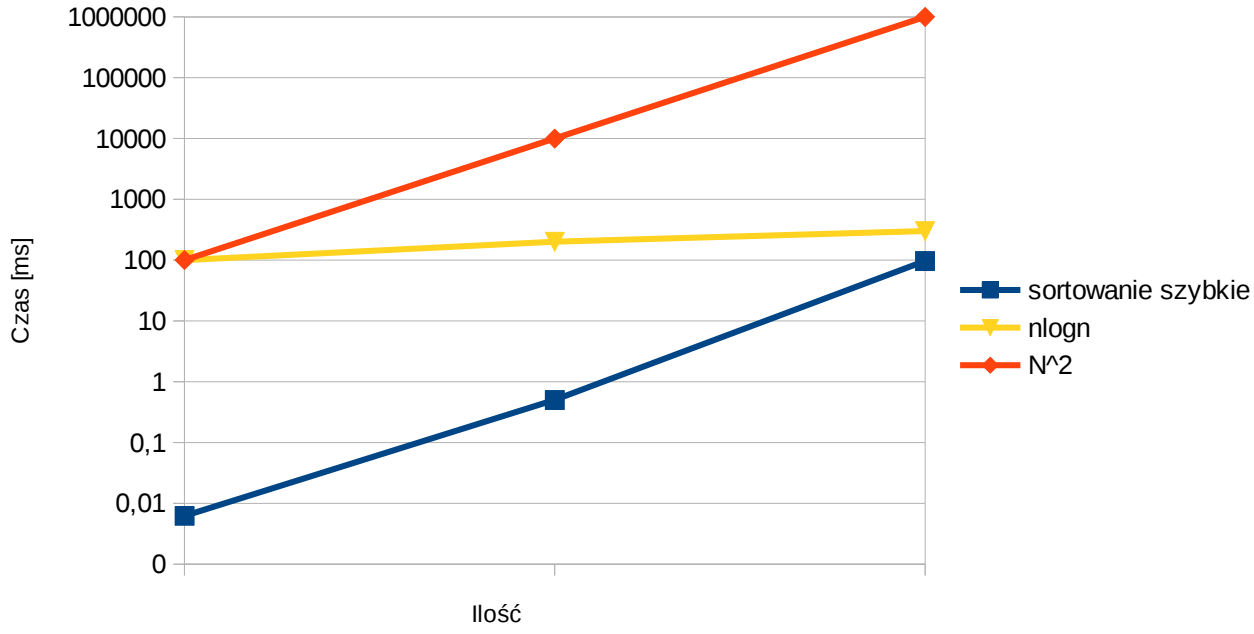
Prosty algorytm korzystający z rekurencyjnej metody „dziel i zwyciężaj”. Jego średnia złożoność obliczeniowa wynosi O(nlogn). Przypadek pesymistyczny – $O(n^2)$ = algorytm zachowuje się jak sortowanie przez wstawianie.

W czasie pisania sprawozdania, algorytm dla 10^7 jeszcze nie skończył pracy. Zastosowano pivota, będącego środkowym elementem listy.

Wyniki:

	10	100	1000	1000000
	0,006	0,354	99,578	
	0,006	0,287	87,278	
	0,005	0,237	122,788	
	0,004	0,47	7,533	
	0,007	0,76	24,214	
	0,011	0,994	109,092	
	0,005	0,601	123,301	
	0,007	0,681	142,606	
	0,006	0,434	189,933	
	0,005	0,201	62,354	
Średnia:	0,0062	0,5019	96,8677	brak danych

Wykres:



Rys. 1. Złożoność czasowa sortowania szybkiego.

Z wykresu widać, że algorytm jest w $O(n^2)$, co się zgadza z teorią.

Sortowanie przez scalanie:

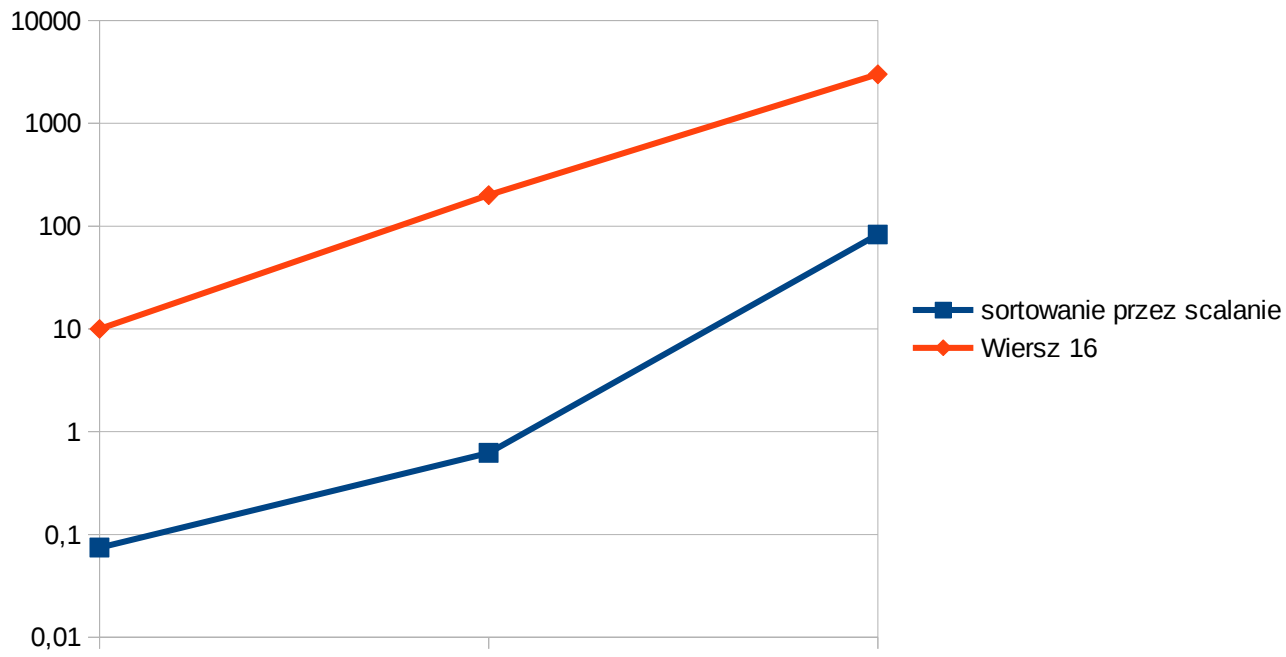
Prosty algorytm korzystający z metody „dziel i zwyciężaj”. Jego złożoność to O(nlogn).

W czasie pisania sprawozdania, algorytm dla 10^7 jeszcze nie skończył pracy.

Wyniki:

10	100	1000	1000000
0,082	0,52	84,308	
0,063	1,46	85,691	
0,063	0,584	82,206	
0,068	0,591	86,109	
0,069	0,566	80,951	
0,058	0,52	76,137	
0,068	0,489	80,906	
0,067	0,494	83,517	
0,077	0,501	85,558	
0,13	0,498	82,748	
0,0745	0,6223	82,8131	brak danych

Wykres:



Rys. 2. Wykres złożoności czasowej algorytmu sortowanie przez scalanie.

Z wykresu widać, że algorytm jest blisko zakładanego czasu działania. Może to wynikać z konkretnej implementacji algorytmu, którą należy zoptymalizować. Może też wynikać z działających aplikacji w tle.