Sprawozdanie z laboratorium nr 6

226543
Tomasz Kaliciak

27 kwietnia 2017

1 Wstęp

Celem ćwiczenia było zaimpementowanie sortowania przez scalanie korzystając z kodu kolegi, zbadanie złożoności obliczeniowej oraz porównanie wyników z wynikami algorytmu szybkiego sortowania.

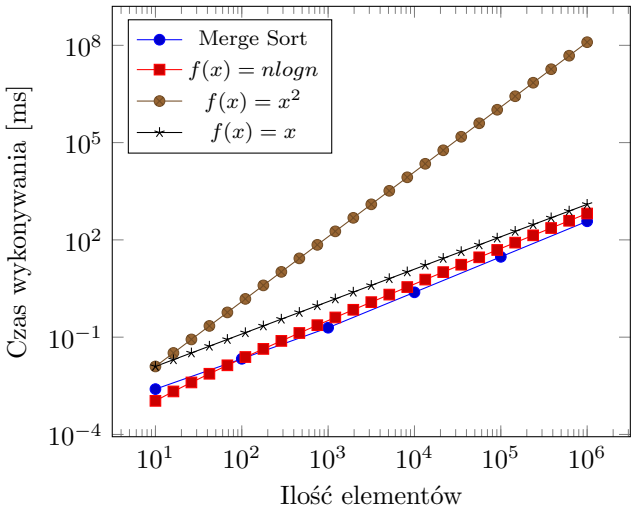
2 Badanie złożoności obliczenioniowej

Pomiary wykonano na tablicy wypełnionej elementami ułożonymi losowo, elementami posortowanymi rosnąco oraz malejąco.

2.1 Tablica z losowo ułożonymi elementami

Tabela 1: Zestawienie czasów sortowania tablicy

	Czas [ms]			
Ilość elementów	QS Pivot losowy	QS Pivot na końcu	QS Pivot na środku	Merge Sort
10^1	0.001	0.00055	0.0005	0.0025
10^2	0.0114	0.01	0.0093	0.0215
10^3	0.13995	0.1195	0.12675	0.19545
10^4	1.74955	1.6195	1.5613	2.4162
10^5	20.1754	18.6207	18.3339	30.00035
10^6	230.796	222.974	211.574	374.947



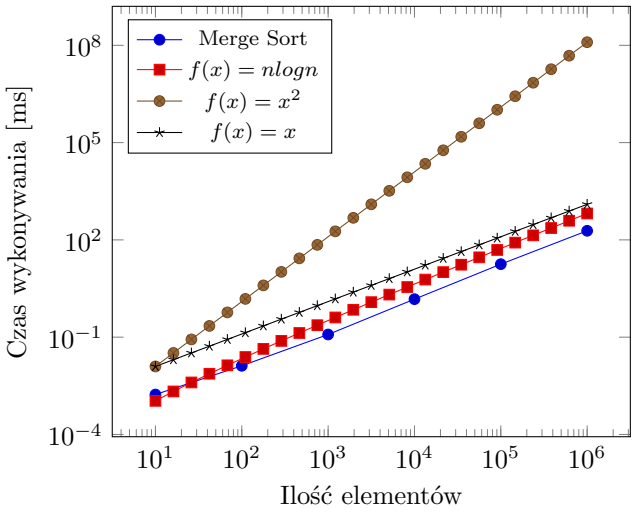
Wykres 1: Czas wykonywania od ilości elementów

W przypadku takiego ułożenia tablicy Merge Sort okazuje się być wolniejszy od Quick Sorta. Z wykresu można odczytać, że złożność obliczeniowa wynosiła w przybliżeniu $O(nlogn)$.

2.2 Tablica z elementami posortowanymi rosnąco

Tabela 2: Zestawienie czasów sortowania tablicy

	Czas [ms]			
Ilość elementów	QS Pivot losowy	QS Pivot na końcu	QS Pivot na środku	Merge Sort
10^1	0.00145	0.00035	0.0002	0.0017
10^2	0.0056	0.0144	0.00265	0.0131
10^3	0.0615	1.2168	0.0277	0.12085
10^4	0.7736	119.916	0.3833	1.4857
10^5	8.31095	11953	4.80975	17.8484
10^6	94.3707	segfault	54.5178	190.823



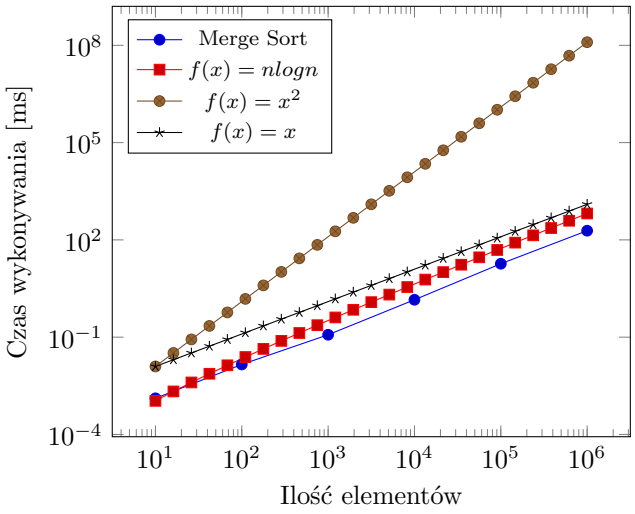
Wykres 2: Czas wykonywania od ilości elementów

Ponownie Merge Sort jest wolniejszy od Quick Sorta. Jednak w tym przypadku tablica jest posortowana w dwukrotnie szybszym czasie. Złożność przypominia coś pomiędzy $O(nlogn)$, a $O(n)$

2.3 Tablica z elementami posortowanymi malejąco

Tabela 3: Zestawienie czasów sortowania tablicy

	Czas [ms]			
Ilość elementów	QS Pivot losowy	QS Pivot na końcu	QS Pivot na środku	Merge Sort
10^1	0.0008	0.0003	0.0003	0.0013
10^2	0.00625	0.01765	0.00375	0.0144
10^3	0.0666	1.23795	0.0408	0.1186
10^4	0.7492	120.877	0.42185	1.4237
10^5	8.4907	12912.9	5.1298	18.3779
10^6	96.1739	segfault	58.3979	191.943



Wykres 3: Czas wykonywania od ilości elementów

Wykresy złożoności obliczeniowej oraz czas wykonywania algorytmu praktycznie nie różni się od wcześniejszego przypadku.

3 Wnioski

- Z przeprowadzonych badań wynika, że złożność obliczeniowa alogrytmu sortowania przez scalanie to $O(nlogn)$
- Algorytm dla badanych przypadków okazał się wolniejszy od Quick Sorta
- W przeciwieństwie do Quick Sorta algorytm posiada stałą złożoność obliczenioną
- Praca w parach pokazuje jak bardzo ważne jest dokumentowanie kodu, nazywanie zmiennych oraz funkcji, prawidlowe projektowanie interfejsu oraz tworzenie funkcji pomocniczych