

Sprawozdanie z laboratorium nr 5

226543

Tomasz Kaliciak

22 kwietnia 2017

1 Wstęp

Celem ćwiczenia było:

- zaimplementowanie algorytmu quicksort
- pomiar złożoności obliczeniowej algorytmu
- zbadanie wpływu wyboru pivotu na przypadek pesymistyczny

2 Badanie złożoności obliczeniowej

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

Przetestowano następujące metody wyboru pivotu:

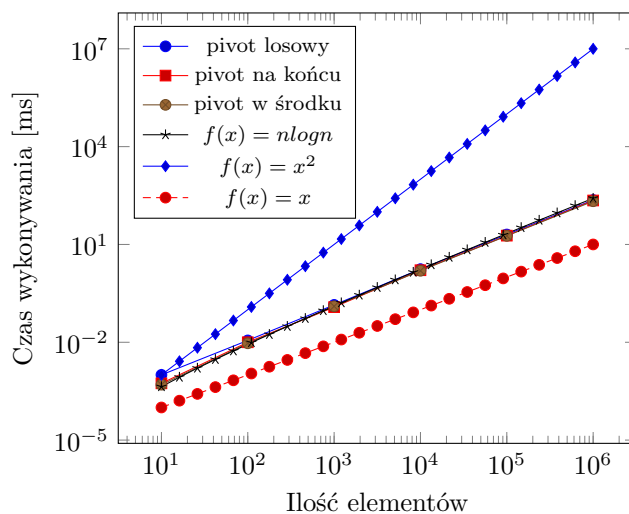
- pivot to losowy element tablicy
- pivot to ostatni element tablicy
- pivot to środkowy element tablicy

Wykonano po 20 pomiarów dla każdego przypadku, wyniki uśredniono.

2.1 Tablica z losowo ułożonymi elementami

Tabela 1: Zestawienie czasów sortowania tablicy

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



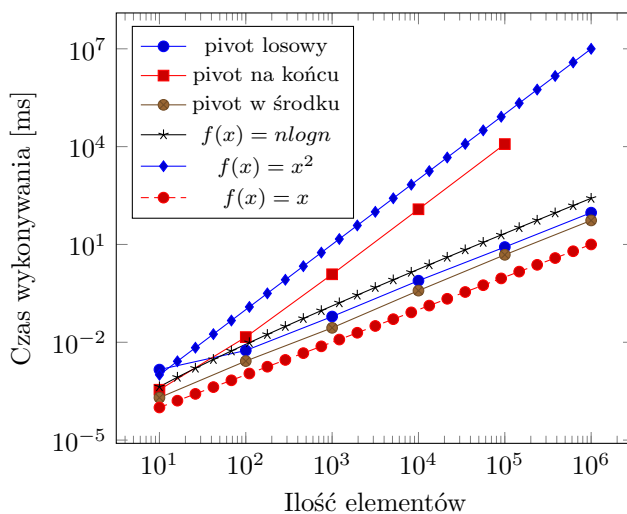
Wykres 1: Czas wykonywania od ilości elementów

Wyniki pomiarów czasów sortowania różnią się nieznacznie. Z wykresu można odczytać, że złożoność obliczeniowa w każdym z przypadków wyboru pivotu wynosiła w przybliżeniu $O(n \log n)$.

2.2 Tablica z losowo ułożonymi posortowanymi rosnąco

Tabela 2: Zestawienie czasów sortowania tablicy

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



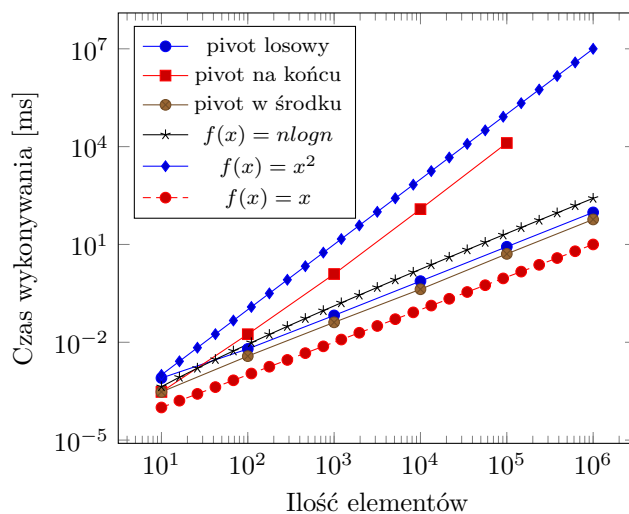
Wykres 2: Czas wykonywania od ilości elementów

Złożoność obliczeniowa w przypadku wybieraniu pivotu jako ostatniego elementu tablicy przypomina funkcję $f(x) = x^2$, natomiast pozostałe dwie metody wyboru pivotu zachowują się jak funkcja $f(x) = n \log n$. Metoda wykorzystująca środkowy element sortuje elementy prawie dwukrotnie szybciej niż metoda wykorzystująca losowy element.

2.3 Tablica z losowo ułożonymi posortowanymi malejąco

Tabela 3: Zestawienie czasów sortowania tablicy

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



Wykres 3: Czas wykonywania od ilości elementów

Wykresy złożoności obliczeniowej praktycznie nie różnią się od wcześniejszego przypadku. Sortowanie zajmuje jednak większą ilość czasu.

3 Wnioski

- Z przeprowadzonych badań wynika, że średnia złożoność obliczeniowa algorytmu szybkiego sortowania to $O(n \log n)$
- W przypadku pesymistycznym złożoność obliczeniowa przypomina $O(n^2)$. Dzieje się tak np. w przypadku wyboru pivotu jako ostatniego elementu w posortowanej tablicy. Każde wywołanie rekurencyjne powoduje wybór na pivot największego lub najmniejszego dostępnego elementu co prowadzi do potrzeby wykonania dużej ilości obliczeń
- Czas sortowania zbioru uporządkowanego rosnąco oraz uporządkowanego malejąco jest praktycznie taki sam
- Teoretycznie rzecz biorąc wybierając losowo pivotu minimalizujemy prawdopodobieństwo zajścia najgorszego przypadku ($O(n^2)$)
- Wybieranie środkowego elementu okazało się najlepszą metodą w przypadku badanych danych wejściowych. Może być to spowodowane tym, że losowanie elementu jest obciążone pewnym kosztem
- Pomimo najlepszych wyników metody wybierania środkowego elementu jako pivot, nie można stwierdzić, że jest to najlepsza metoda wyboru pivotu, ponieważ dokonano zbyt mało pomiarów
- Na czas wykonania sortowania ma wpływ zarówno wybór pivotu jak i dane wejściowe