Sprawozdanie laboratorium 5 - quicksort

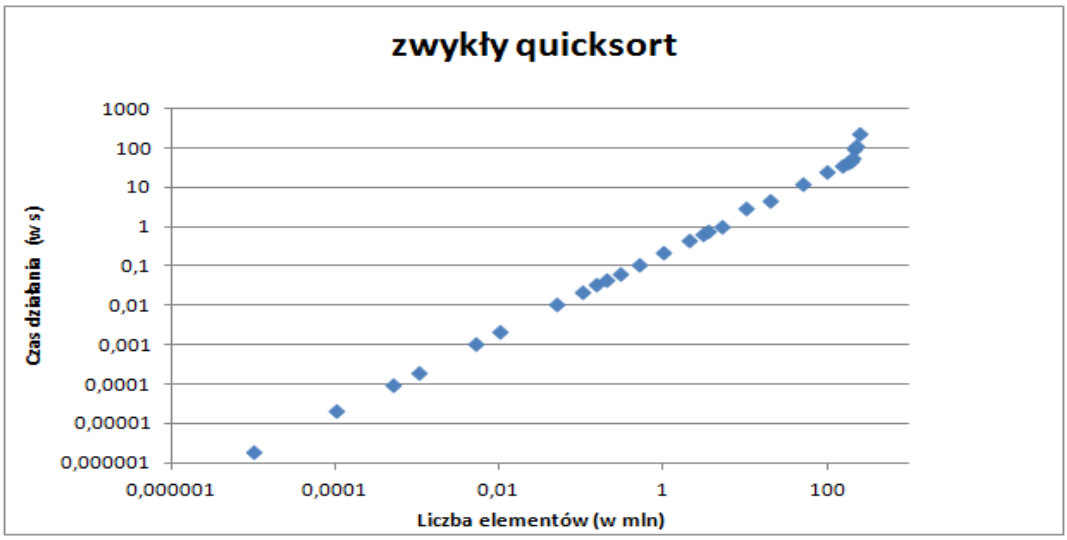
200439

31 marca 2014

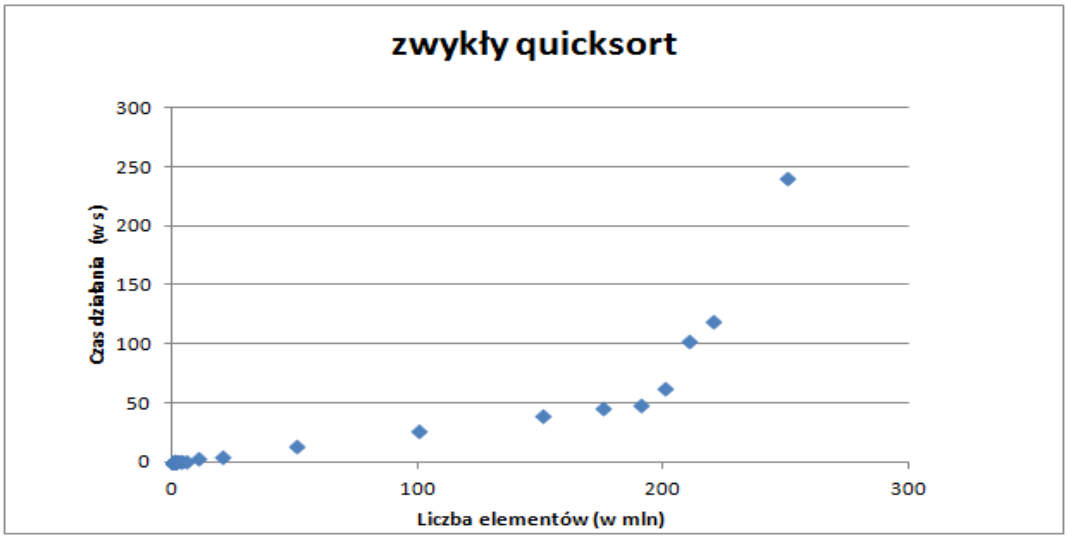
Sortowanie szybkie (quicksort) jest jednym z najszybciej działających algorytmów sortowania. W optymalnym przypadku (liczby ustawione początkowo w sposób losowy) jego złożoność obliczeniowa i czas wykonania to O(NlogN). Jednak dla szczególnego przypadku, gdy elementy przed wykonaniem algorytmu są już posortowane, złożoność obliczeniowa quicksort staje się złożonością kwadratową. Można to zmienić, stosując modyfikację algorytmu szybkiego sortowania.

Złożoność algorytmu zależy od wyboru elementu rozdzielającego (pivot). Najprostszą metodą jego wyboru jest wybieranie za każdym razem elementu skrajnego tablicy. Gdy tablica jednak jest już posortowana, każdy podział tej tablicy tworzy obszary o rozmiarach N-1 i 1. Żeby uniknąć takiej sytuacji, stosuje się randomizację wyboru elementu rozdzielającego. Wtedy prawdopodobieństwo zajścia najgorszego przypadku sprowadza się do wartości zaniedbywalnie małych.

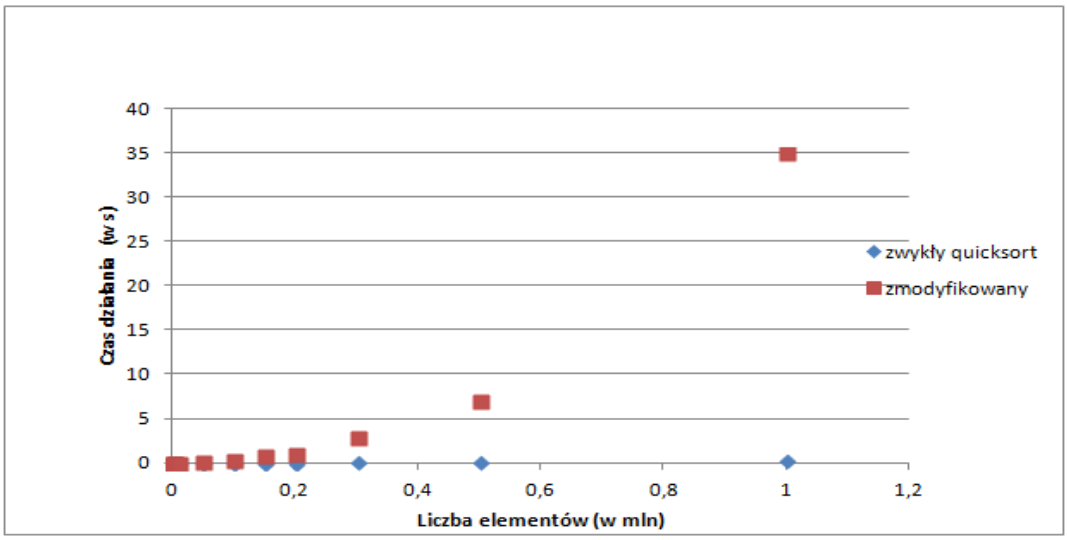
Na poniższym wykresie można zauważyć, że czas działania zwykłego, niemodyfikowanego algorytmu quicksort w przypadku liczb pomieszanych jest zdecydowanie lepsza od wersji zmodyfikowanej.



Wykres 1. Czasy sortowania tablicy elementów rozmieszczonych w kolejności losowej zwykłym algorytmem quicksort. Osie wyskalowane logarytmicznie. Zauważyć można, że złożoność tego algorytmu jest logarytmiczna (O(NlogN)).

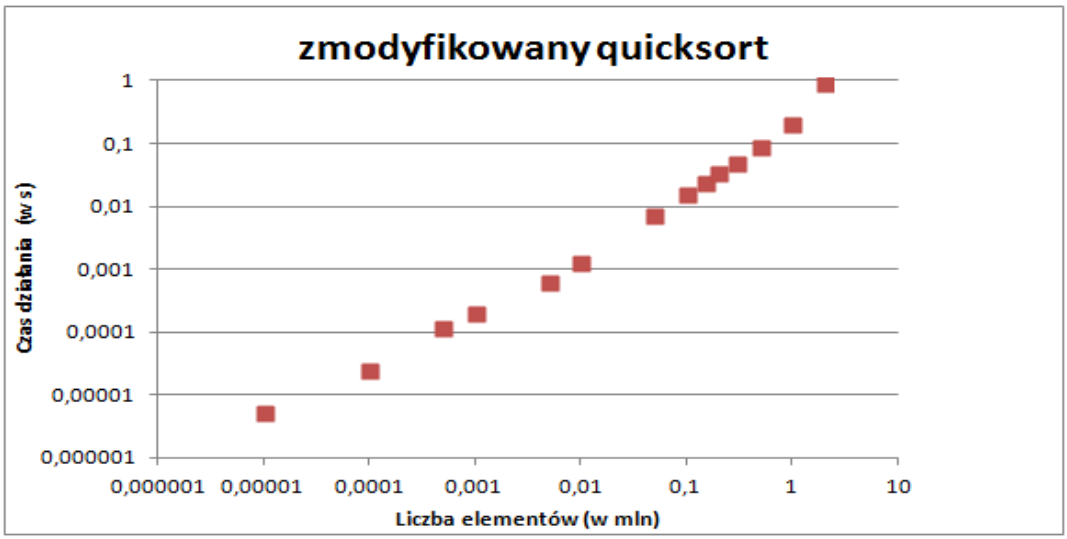


Wykres 2. Czasy sortowania tablicy elementów rozmieszczonych w kolejności losowej zwykłym algorytmem quicksort. Osie wyskalowane liniowo.

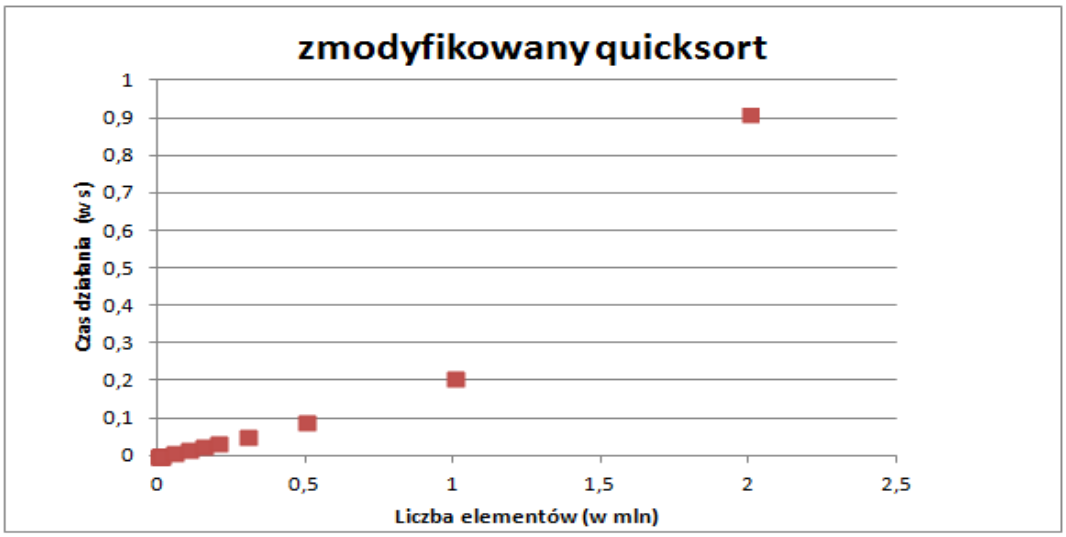


Wykres 3. Porównanie czasów sortowania tablicy elementów rozmieszczonych w kolejności losowej zwykłym algorytmem quicksort oraz wersją zmodyfikowaną. Osie wyskalowane liniowo.

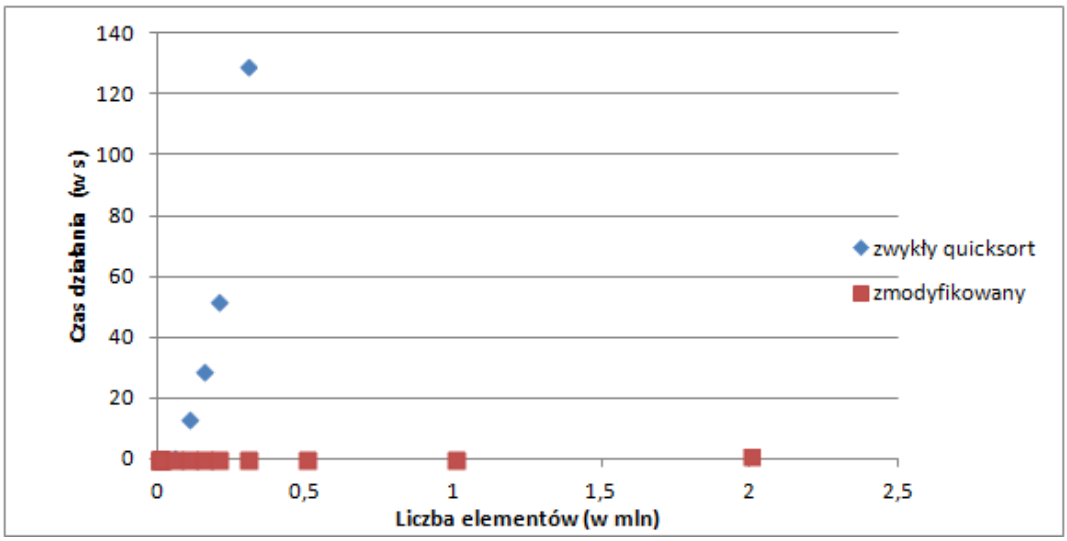
Kiedy jednak elementy tablicy będą już posortowane (rosnąco lub malejąco), zwykły algorytm sortowania szybkiego staje się praktycznie nie do użytku dla większych ilości danych. Jego złożoność obliczeniowa zwiększa się do kwadratowej, podczas gdy czas sortowania algorytmem z losowym pivot'em nadal ma złożoność logarytmiczną. Widać to dobrze na poniższych wykresach.



Wykres 4. Czasy sortowania tablicy elementów rozmieszczonych w uporządkowanej, rosnącej kolejności zmodyfikowanym algorytmem quicksort. Osie wyskalowane logarytmicznie. Zauważyć można, że złożoność tego algorytmu jest logarytmiczna (O(NlogN)).



Wykres 5. Czasy sortowania tablicy elementów rozmieszczonych w uporządkowanej, rosnącej kolejności zmodyfikowanym algorytmem quicksort. Osie wyskalowane liniowo.



Wykres 6. Porównanie czasów sortowania tablicy elementów rozmieszczonych w uporządkowanej, rosnącej kolejności zwykłym algorytmem quicksort oraz wersją zmodyfikowaną. Osie wyskalowane liniowo.

Przedstawione przykłady sortowania tablic pokazują, że czas wykonania algorytmów zależy od początkowego układu elementów. Każdy z wariantów sortowania szybkiego ma swoje wady i zalety. Zwykły quicksort sprawdza się lepiej przy sortowaniu liczb losowych, natomiast jego złożoność staje się kwadratowa dla liczb uprzednio posortowanych. Algorytm z randomowym pivot'em ma większy czas działania dla liczb losowych, lecz nie osiąga złożoności kwadratowej dla posortowanych wcześniej liczb.

Istnieją także inne rodzaje usprawnień algorytmu sortowania szybkiego, które jednak nie zostały zaimplementowane w tym ćwiczeniu.