

# Różne przypadki złożoności obliczeniowej algorytmu Quicksort

Monika Litwin 200586

23 kwietnia 2014

## Trochę teorii

### Przypadek optymistyczny

Teoretycznie takowy występuje, gdy sortowany jest już uporządkowany zbiór (z niewielką ilością elementów nie na swoich miejscach).  
Wtedy złożoność wynosi  $O(n \log n)$

### Przypadek typowy

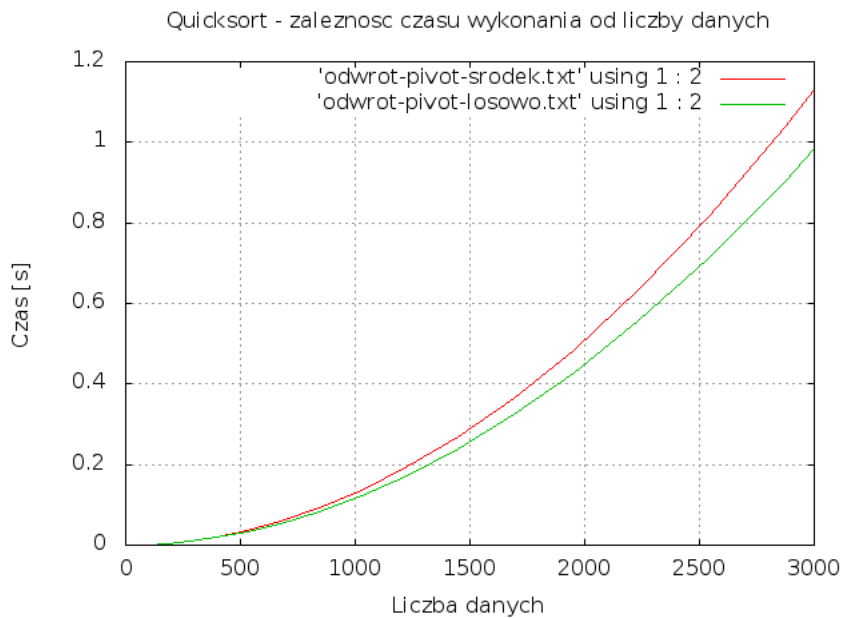
Zbiór poddany algorytmowi sortowania posiada losowo rozłożone elementy.  
Złożoność wynosi  $O(n \log n)$

### Przypadek pesymistyczny

Występuje dla zbiorów o elementach posortowanych odwrotnie.  
Złożoność wynosi  $O(n^2)$

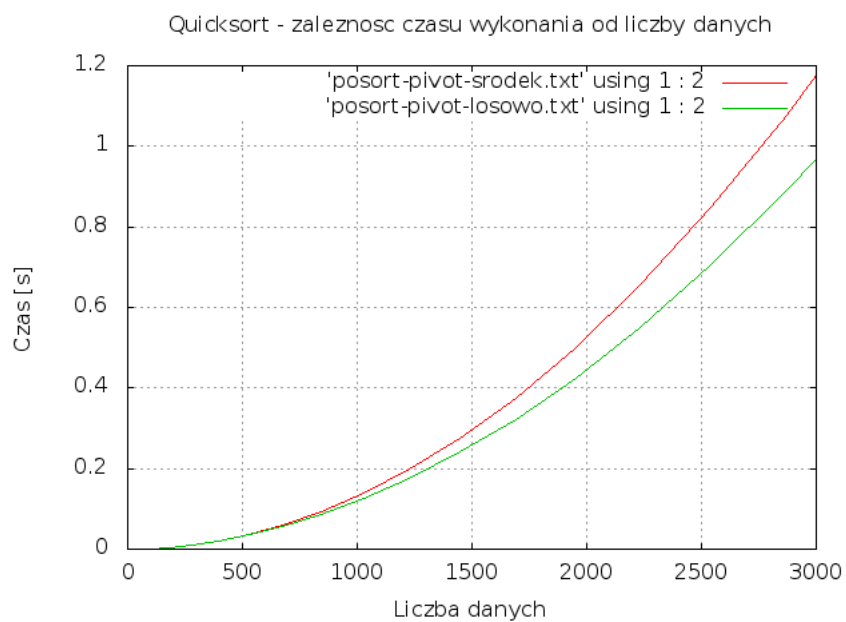
### **Sortowanie zbioru odwrotnie posortowanych elementów**

Poniżej zamieszczony jest wykres pomiarów czasu dla algorytmu, któremu podano wcześniej posortowany w odwrotnej kolejności wektor. Testy wykonano dla przypadku, gdy pivot wybierany jest w sposób losowy lub zawsze jest elementem środkowym. Czas sortowania jest krótszy, gdy pivot jest losowy.



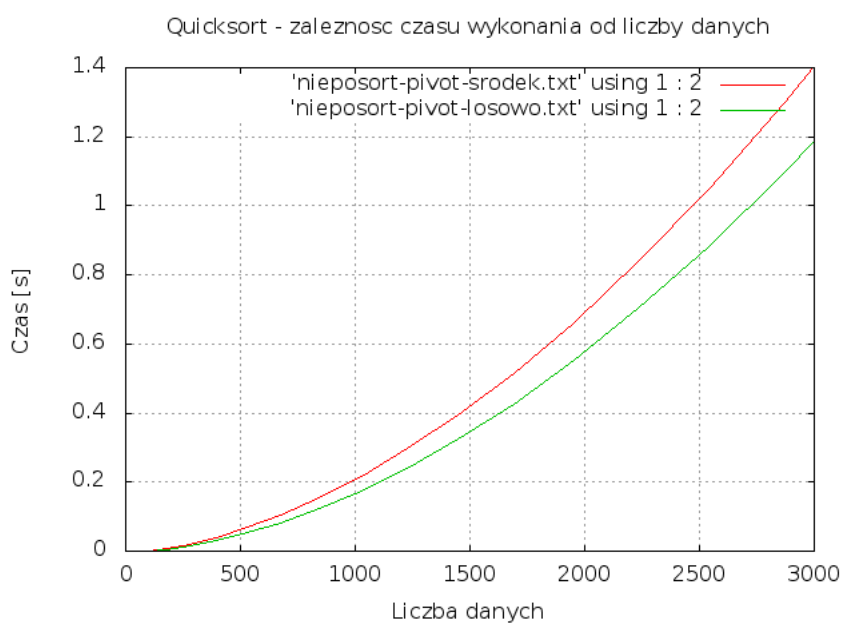
### Sortowanie zbioru posortowanych elementów

W kolejnym teście porównuję wpływ wyboru pivotu na czas sortowania dla przetwarzania posortowanego już wektora. Tutaj również algorytm wykonywał się szybciej, gdy pivot wybierany był losowo.



### Sortowanie zbioru nieposortowanych elementów

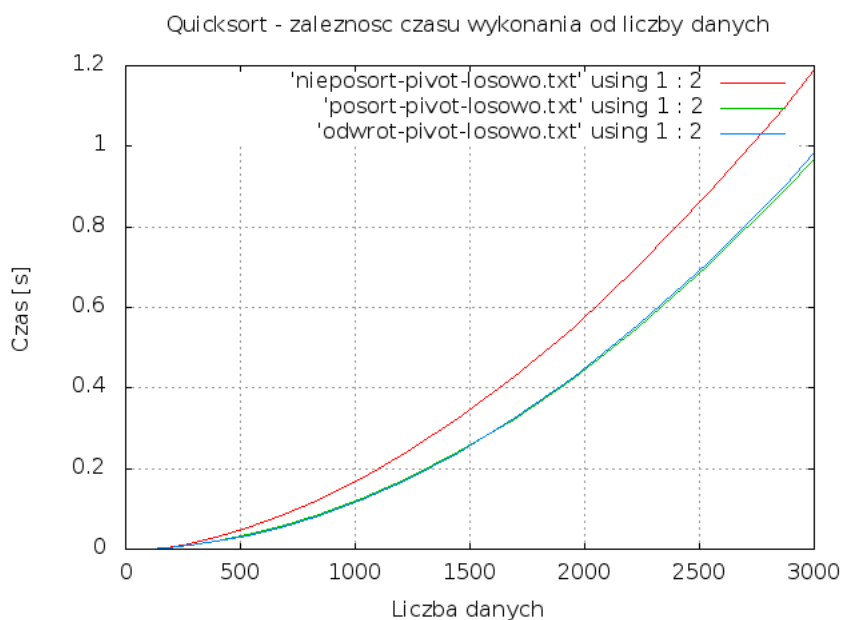
Gdy sortujemy zbiór nieuporządkowanych elementów sytuacja jest identyczna, jak w poprzednich dwóch przypadkach - algorytm wykonuje się szybciej, kiedy pivot wybierany jest losowo.



### Sortowanie elementów z losowo wybranym pivotem

Tutaj obserwujemy szybkość wykonania algorytmu Quicksort z losowym pivotem dla zbiorów: nieposortowanego, posortowanego oraz posortowanego odwrotnie.

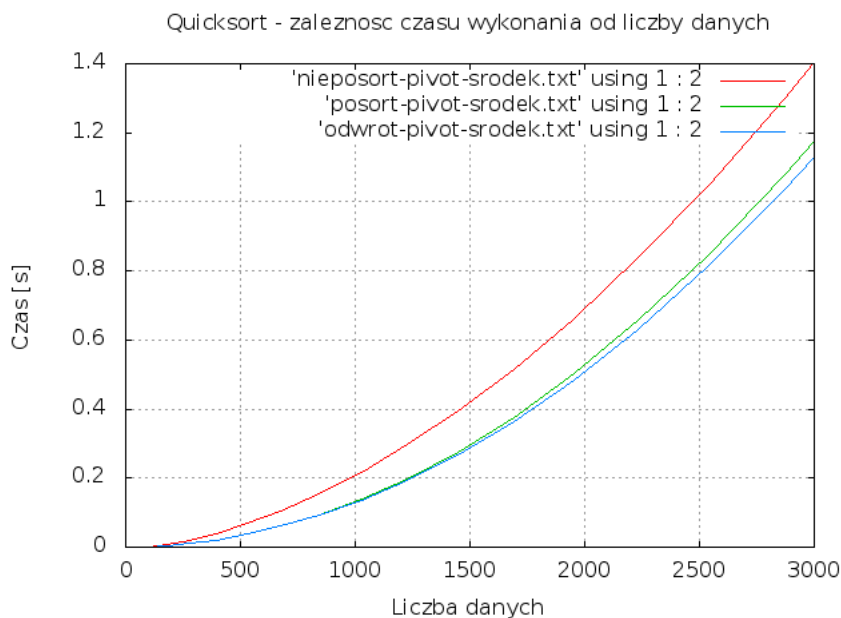
Zdecydowanie najwięcej czasu potrzeba było na posortowanie elementów wcześniej nieposortowanych. Pozostałe przypadki są bardzo zbliżone czasowo do siebie.



### Sortowanie elementów z pivotem po środku

Na poniższym wykresie obserwujemy sytuację niemal identyczną, jak na poprzednim.

Najdłużej sortuje się zbiór nieposortowany, pozostałe - podobnie czasowo.



### Wnioski

Można zaobserwować, że losowe dobieranie pivotu faktycznie usprawniło działanie algorytmu. Widać wyraźną różnicę na wykresach obrazujących czas sortowania (pierwsze trzy).

Jeśli chodzi o porównanie przedstawionej teorii z praktycznymi wynikami, to nie zaobserwowałam przypadku pesymistycznego. Miał on wystąpić przy sortowaniu odwrotnie posortowanych elementów. W rzeczywistości czas sortowania takiego zbioru jest bardzo zbliżony do sortowania elementów posortowanych we właściwym porządku.

Zarówno, gdy pivot jest dobierany losowo, jak i wtedy gdy zawsze jest elementem środkowym. Wyraźnie za to odbiega przypadek sortowania nieuporządkowanego zbioru - czas jest dłuższy.