

Różne przypadki złożoności obliczeniowej algorytmu Quicksort

Monika Litwin 200586

28 marca 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 $O(2n \log n)$

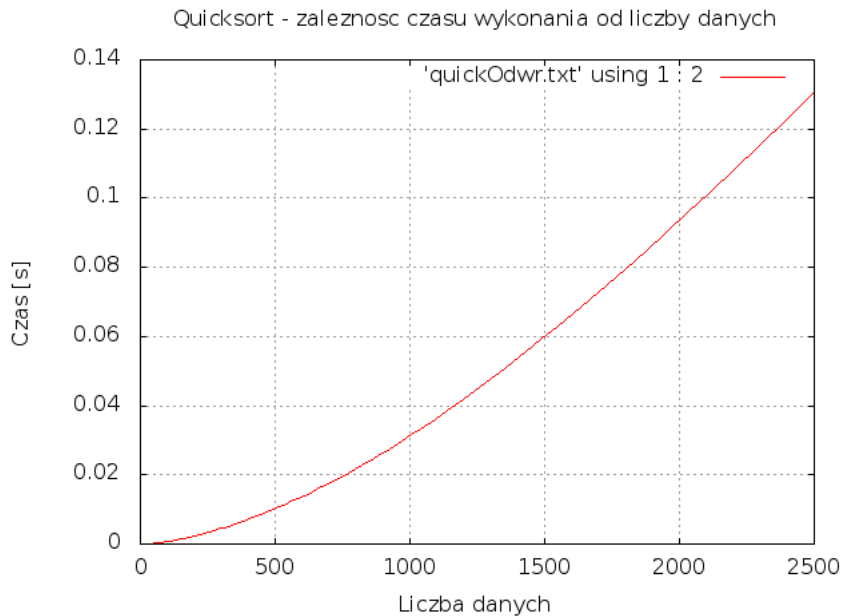
Przypadek pesymistyczny

Występuje dla zbiorów o elementach posortowanych odwrotnie.

Złożoność wynosi $O(\frac{n^2}{2})$

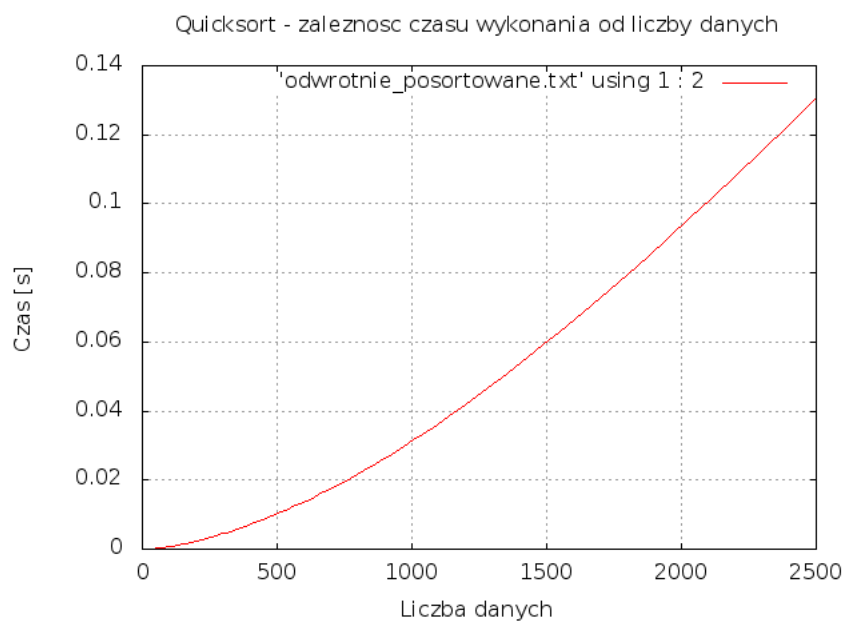
Sortowanie zbioru odwrotnie posortowanych elementów

Poniżej zamieszczony jest wykres pomiarów czasu dla algorytmu, któremu podano wcześniej posortowany w odwrotnej kolejności wektor. W testach mojego algorytmu sortowania Quick, najszybszy okazał się właśnie ten przypadek.



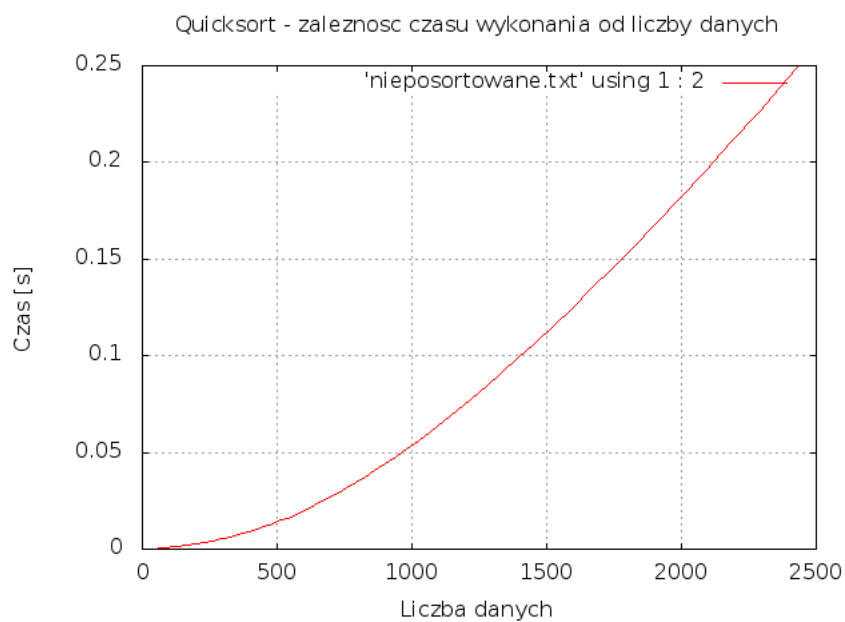
Sortowanie zbioru posortowanych elementów

Kolejnym jeśli chodzi o szybkość wykonania okazał się być algorytm, w którym sortujemy posortowany już wektor. Jego czas wykonania nie różni się znacznie od wcześniejszego przypadku.



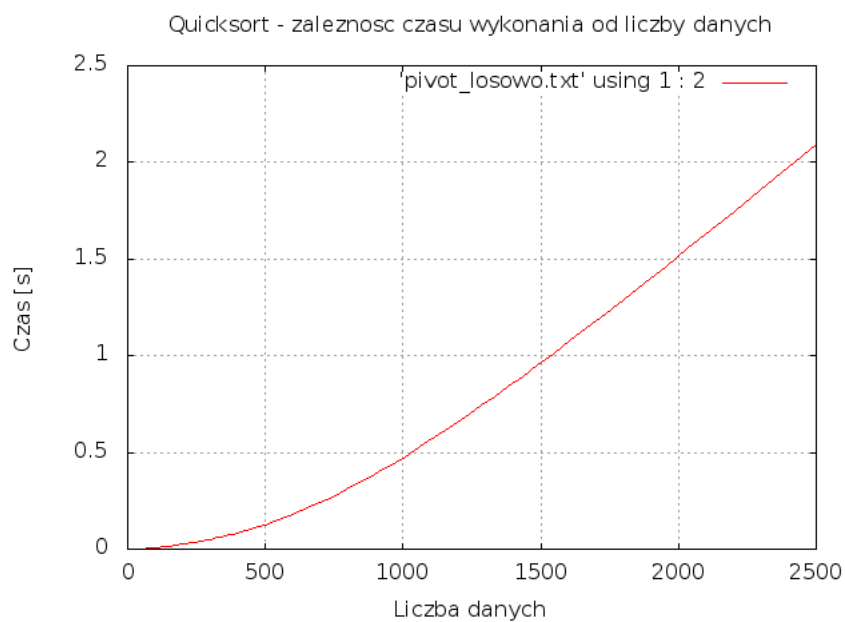
Sortowanie zbioru rozłożonych losowo elementów

Jeszcze wolniej wykonuje się nasz algorytm Quicksort, gdy podamy do niego wektor z nieuporządkowanymi elementami. Jednak również jest to niewielkie odchylenie w stosunku do wcześniejszych przypadków.



Sortowanie zbioru nieuporządkowanego z losowo wybranym pivotem

W testach ten przypadek sortowania szybkiego wypadł zdecydowanie najgorzej. Okazał się stanowczo najwolniejszy. Znacząco odbiega swoją złożonością od wcześniej prezentowanych.



Porównanie wszystkich przypadków

Po testach okazało się, że teoria niewiele ma wspólnego z otrzymanymi wynikami pomiarów. Sortowanie odwrotnie posortowanych elementów - przypadek, który miał być najwolniejszy okazał się najszybszy. Algorytm, do którego podawałam prawidłowo posortowane dane plasuje się pośrodku, zamiast, zgodnie z przypuszczeniami - na czele. Najwolniejszy, i to bardzo wyraźnie, jest przypadek, gdzie element osiowy wybierany był losowo, co według założeń miało usprawnić działanie algorytmu, zmniejszając prawdopodobieństwo wystąpienia opcji pesymistycznej. Wynika to prawdopodobnie z różnic w sposobie implementacji algorytmu Quicksort, przyjętego sposobu mierzenia czasu lub ewentualnego błędu w trakcie przeprowadzania pomiarów.

