

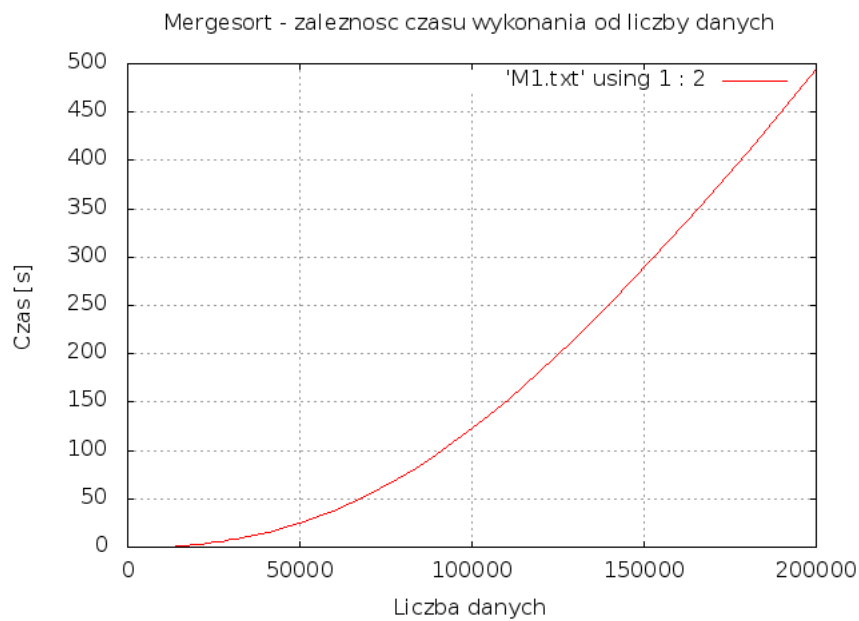
Czas realizacji algorytmów sortowania.  
Mergesort, Heapsort, Quicksort.

Monika Litwin 200586

22 marca 2014

## Mergesort

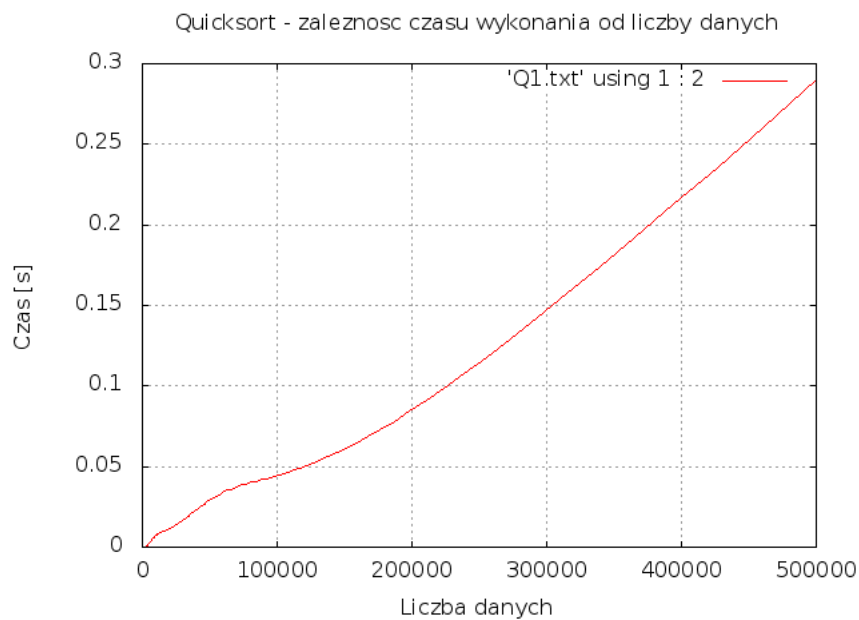
Algorytmu sortowania Mergesort należy do algorytmów szybkich, czyli posiadających klasę czasowej złożoności obliczeniowej równą  $O(n \log n)$  lub lepszą. Mergesort w teorii cechuje się właśnie złożonością  $O(n \log n)$ . Patrząc na wykres można uznać, że tak właśnie zachowuje się zaimplementowane przeze mnie w programie sortowanie merge.



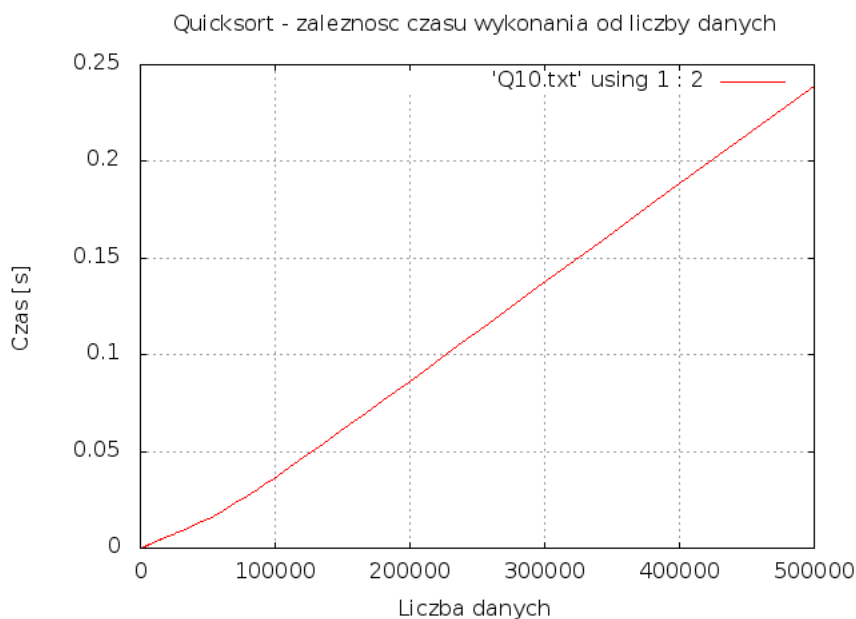
W testach ten typ sortowania wypadł zdecydowanie najgorzej. Okazał się stanowczo najwolniejszy.

## Quicksort

Quicksort to prawdopodobnie najczęściej obecnie stosowany algorytm sortowania. Dla średniego przypadku jego złożoność obliczeniowa wynosi  $O(n \log n)$ . Zbliżoną do niej złożoność obliczeniową uzyskano w testach, zobrazowanych poniższym wykresem.



Algorytm ten posiada przypadek optymistyczny i pesymistyczny, których złożoność obliczeniowa znacznie się różni. Przypadek pesymistyczny to taki, kiedy sortujemy uporządkowaną już tablicę. Wówczas podtablica nie jest praktycznie dzielona, ale odpada od niej jedynie element osiowy, a pozostałą część podtablicy należy dzielić dalej. Prawdopodobnie właśnie taki przypadek otrzymaliśmy, gdy zadane było 10 powtórzeń algorytmu. Tylko za pierwszym razem wektor faktycznie sortowano, pozostałe 9 razy sortowany był, już wcześniej uporządkowany wektor. Obraz tego mamy na wykresie poniżej.



W odróżnieniu od algorytmu sortowania merge, tutaj przy podziale danych na podtablice, dąży się do tego, aby podział na podtablice był jak najbardziej równomierny (co powoduje znaczne zwiększenie złożoności w przypadku niekorzystnym). W sortowaniu merge, podział tablicy jest maksymalnie uproszczony - zawsze jest dzielona na połowy, bez wnikania w ułożenie danych, Faktyczne sortowanie ma miejsce w trakcie scalania tablic. W Quicksort jest odwrotnie - sortowanie ma miejsce w trakcie podziału, a łączenie jest automatyczne.

## Heapsort

Sortowanie typu heap również zalicza się do algorytmów szybkich. Cechuje się złożonością liniowo logarytmiczną. Jest nieco wolniejszy niż quick, ale w odróżnieniu od tego, wykazuje bardzo małą wrażliwość na postać danych i utrzymuje stałą złożoność czasową.

