

“Quick Sort, Merge Sort - algorytmy sortowania”

Martyna Bandura

23 marca 2014

1. Czym jest sortowanie?

Jeden z podstawowych problemów informatyki. Polega na uporządkowaniu zbioru danych względem pewnych cech charakterystycznych każdego elementu tego zbioru. Szczególnym przypadkiem jest sortowanie względem wartości każdego elementu, np. sortowanie liczb, słów itp.

Algorytmy sortowania są stosowane w celu uporządkowania danych, umożliwienia stosowania wydajniejszych algorytmów (np. wyszukiwania) i prezentacji danych w sposób czytelniejszy dla człowieka.

2. Przykładowe rodzaje sortowania:

- sortowanie bąbelkowe;
- sortowanie przez wstawianie;
- sortowanie przez scalanie;
- sortowanie Shella;
- sortowanie szybkie;
- sortowanie introspektywne;
- sortowanie przez kopcowanie.

3. Opis wykorzystanych algorytmów:

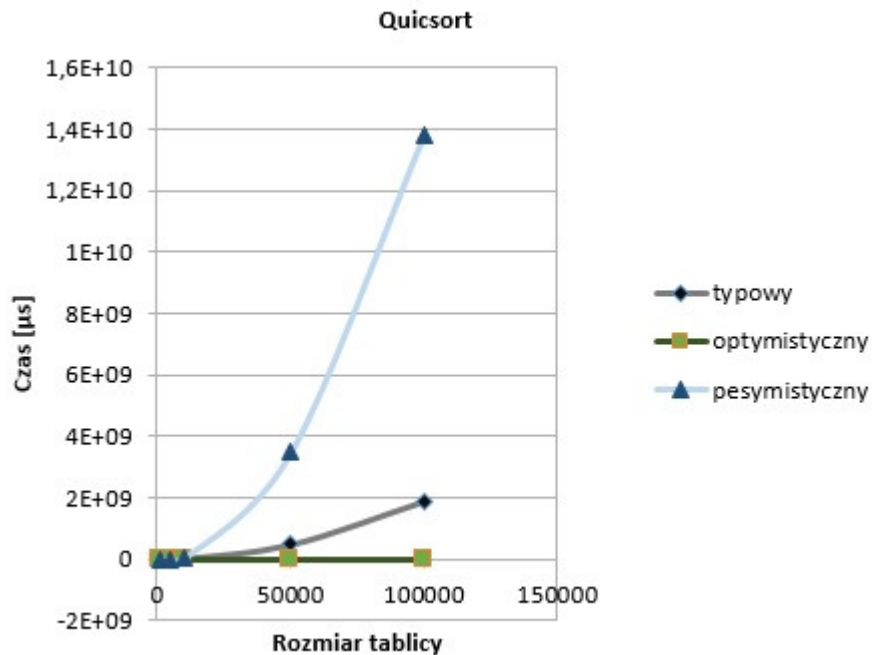
- **Quick Sort:**
 - *Sortowanie QuickSort zostało wynalezione przez C.A.R. Hoare'a. Jest to jeden z najpopularniejszych algorytmów sortowania. Wpłynęły na to dwie rzeczy. Po pierwsze jest ono bardzo szybkie (jak sama nazwa wskazuje), a po drugie - proste do wytłumaczenia i implementacji. Pesymistyczny czas jego działania wynosi $O(n^2)$, a średni $O(n \cdot \lg(n))$. Mimo tego w praktyce jest to najszybszy algorytm sortowania dużych tablic danych.*
 - *Sortowanie szybkie opiera się na technice "dziel i zwyciężaj". Wejściowa tablica jest dzielona (po przestawieniu niektórych z jej elementów) na dwie mniejsze. Każdy element pierwszej tablicy nie jest większy niż każdy element drugiej tablicy. Liczbę, według której wykonuje się podziału to najczęściej pierwszy element tablicy. Następnie dla tych dwóch podtablic wywoływany jest rekurencyjnie ten sam algorytm. Wywołania rekurencyjne kończą się aż któraś z kolejnych podtablic będzie zawierała tylko jeden element. QuickSort działa w miejscu.*

- Merge Sort:

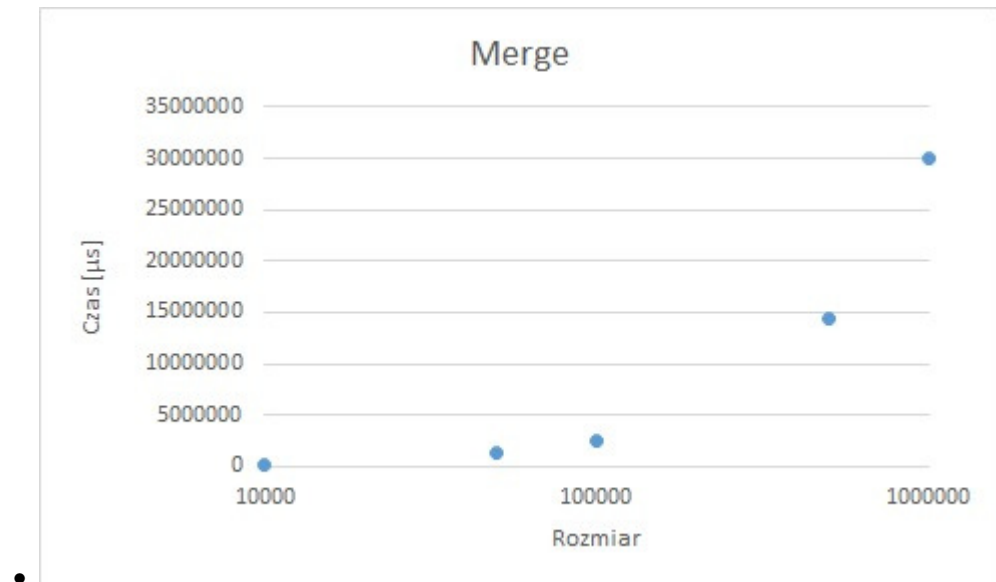
- Bazuje ono na zasadzie "dziel i rządź" - czyli na podzieleniu problemu na mniejsze części, co znacznie ułatwia ich rozwiązywanie. Sortowanie przez scalanie polega na sukcesywnym połowieniu szeregu danych wejściowych do momentu pozostania z jednym elementem w zbiorze. Taki zbiór, ze względu na jego liczebność, bez wahania możemy nazwać posortowanym. Następnym krokiem jest scalenie takich jednoelementowych zbiorów w jeden, większy. Zgodnie z tą regułą "cofamy się", aż wszystkie takie podzbiory scalimy w całość, uzyskując posortowany ciąg wejściowy.

4. Testy wydajności:

- Quick Sort - Dane przedstawiają algorytm sprawdzający działanie algorytmu w skrajnych przypadkach.



- Merge Sort



Wnioski:

- Algorytmy Merge Sort i Quick Sort mają złożoność $O(\log_2 n)$.
- Tak jak w opisie algorytmu Quick Sort okazał się najszybszy.
- Otrzymane czasy sortowania są proporcjonalne do iloczynu $n \log_2 n$, zatem złożoność obliczeniowa algorytmu sortowania szybkiego jest równa $O(n \log n)$.
- Złożoność obliczeniowa przypadku pesymistycznego wynosi $O(n^2)$.
- Czas sortowania zbioru nieuporządkowanego jest wyraźnie dłuższy od czasów sortowania zbiorów uporządkowanych częściowo.