

Sprawozdanie z laboratorium nr 5 „Sortowania”

Karolina Morawska

30.03.2014

1 Wstęp

Celem ćwiczenia było przetestowanie różnych algorytmów sortowania. Wybrane przeze mnie :

- Introspektywne
- Merge sort - przez scalanie
- Sortowanie szybkie

2 Opis wybranych przeze mnie algorytmów :

Wybrano sortowanie szybkie którego złożoność obliczeniowa wynosi $O(n \log n)$. Jest to sortowanie wydajne oraz często używane ze względu na prostotę implementacji i szybkość. Następnym sortowaniem jest mergesort-rekurencyjny algorytm sortowania danych który działa w taki oto sposób:

- dzieli zestaw danych na dwie równe części
- stosuje sortowanie przez scalanie dla każdej z nich oddzielnie, chyba że pozostał już tylko jeden element;
- łączy posortowane podciągi w jeden.

Jego złożoność obliczeniowa jest taka sama jak w przypadku quicksort. Ostatnie sortowanie to introspektywne. W przypadku ogólnym, a więc również w najgorszym, algorytm Sortowania Introspektywnego posiada złożoność obliczeniową taką samą jak dwa poprzednie. W najgorszym przypadku algorytm wykonuje najpierw rekurencyjne wywołanie, takie jak w Sortowaniu Szybkim, a następnie dla pozostałego podzbioru wywołuje procedurę Heap Sort. Jest on algorytmem sortującym w miejscu.

2.1 Introspektywne.

N	Czas [ms]
10 000	1,26
50 000	5,169
100 000	21,1
500 000	74,044
1000000,00	123,902

Rysunek 1: Tabela rozmiaru problemu od czasu dla sortowania introspektywnego.



Rysunek 2: Wykres zależności rozmiaru problemu od czasu działania algorytmu dla sortowania introspektywnego.

2.2 Merge sort.

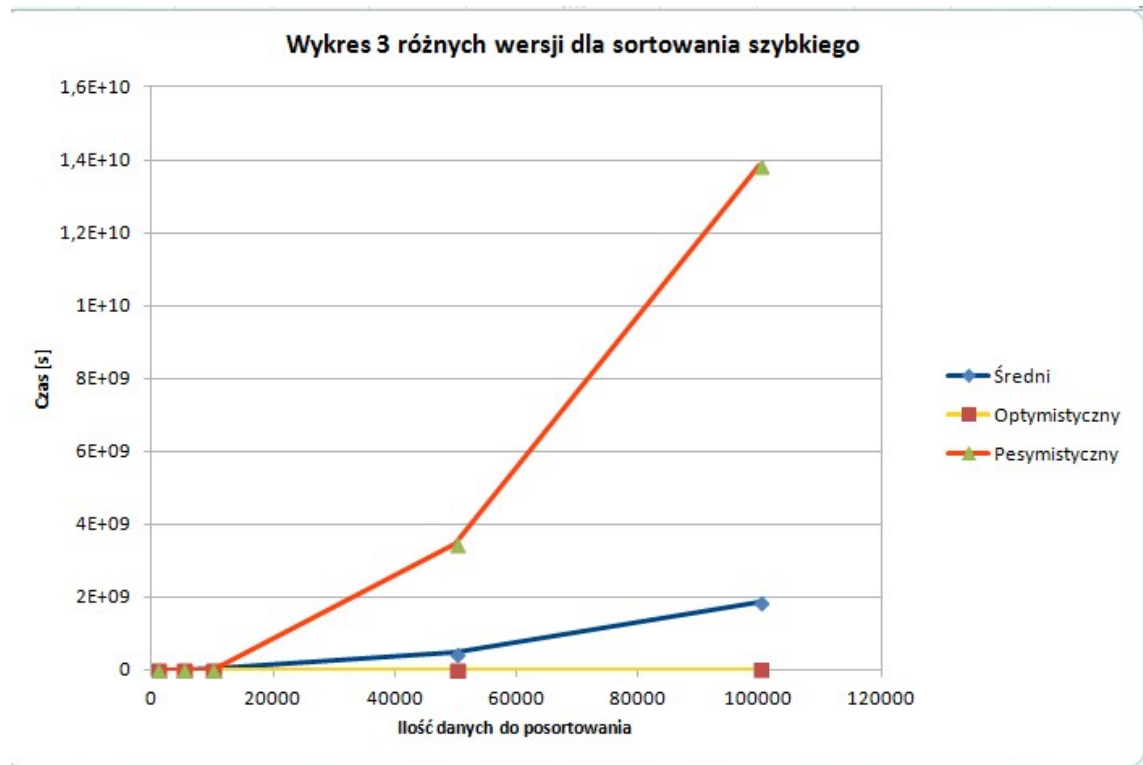
N	Czas [ms]
10 000	1,864
50 000	10,56
100 000	21,198
500 000	81,772
1000000,00	133,421

Rysunek 3: Tabela rozmiaru problemu od czasu działania algorytmu dla sortowania merge czyli przez scalanie.



Rysunek 4: Wykres zależności rozmiaru problemu od czasu działania algorytmu dla sortowania merge czyli przez scalanie.

2.3 Sortowanie szybkie.



Rysunek 5: Wykres zależności rozmiaru problemu od czasu działania algorytmu dla sortowania szybkiego - 3 różne przypadki

3 Wnioski:

- Czas sortowania dla zbiorów nieuporządkowanych jest dużo większy od czasu sortowania dla zbiorów uporządkowanych.
- Optymistyczny przypadek sortowania zachodzi tylko i wyłącznie wtedy gdy kluczem podziału jest mediana z sortowanego zbioru. Podział daje nam równe zbiory.
- W przypadku przeciętnym, gdzie rozkład prawdopodobieństwa wyboru elementu jest taki sam złożoność jest zaledwie wyższa o 39
- Przypadek pesemistyczny zachodzi wtedy, gdy każdy podział jest skrajnie nierównomierny. tj. generuje jedno podzadanie puste, a drugie o 1 mniejsze niż całe zadanie.
- Wszystkie otrzymane czasy sortowania są proporcjonalne do iloczynu $n \log_2 n$, możemy zauważyć zatem, iż klasa złożoności obliczeniowej algorytmu sortowania szybkiego jest równa $O(n \log n)$.