

Projektowanie algorytmów i metody sztucznej inteligencji

Paulina Szczerbak

termin laboratorium: 4.04.2016

1 Temat laboratorium:

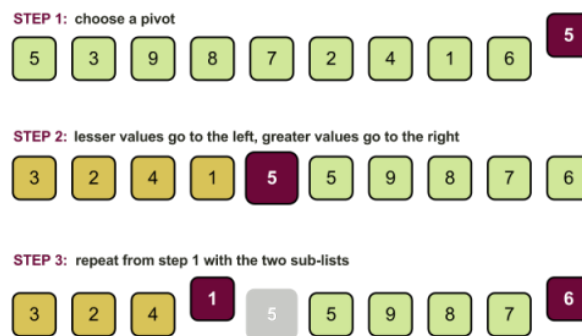
Praca na kodzie kolegi - implementacja sortowania szybkiego i sortowania przez scalanie, wykorzystując wcześniej zaimplementowaną listę.

2 Implementacja:

W tym celu została wykorzystana dwukierunkowa lista oparta na węzłach. Lista dla przypadku średniego wypełniana jest losowymi liczbami z przedziału $[1; \text{rozmiarListy}]$. W sortowaniu szybkim pivot przyjmowany jest jako ostatni węzeł, dla przypadku pesymistycznego sortowana jest posortowana już lista.

3 Sortowanie szybkie (quicksort):

Idea sortowania przedstawiona jest na Rysunku 1.



Rysunek 1: Idea sortowania szybkiego.

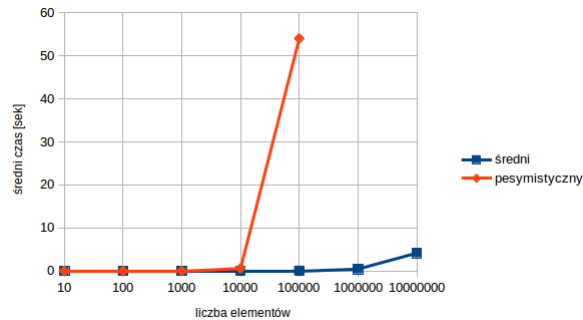
3.1 Przypadek średni:

Liczba danych	10^1	10^2	10^3	10^4	10^5	10^6	10^7
Czas [sek]	0,000000950	0,00003290	0,00015187	0,00181699	0,02230000	0,28084898	4,8199

3.2 Przypadek pesymistyczny:

Liczba danych	10^1	10^2	10^3	10^4	10^5
Czas [sek]	0,0000031	0,00011396	0,00590801	0,59724689	54,00412369

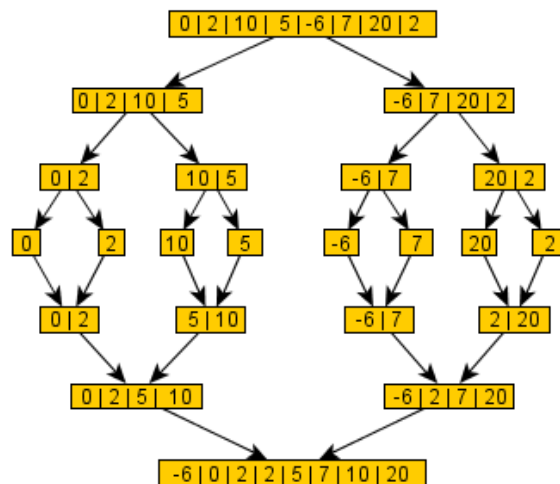
Czas sortowania dla przypadku średniego i pesymistycznego jest przedstawiony na Rysunku 2.



Rysunek 2: Czas sortowania dla przypadku średniego i pesymistycznego

4 Sortowanie przez scalanie (mergesort):

Idea sortowania przedstawiona jest na Rysunku 3.

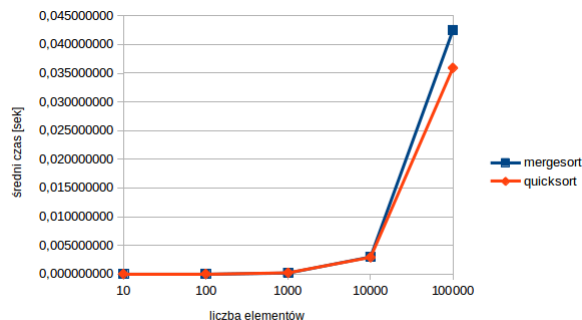


Rysunek 3: Idea sortowania przez scalanie.

4.1 Przypadek średni:

Liczba danych	10^1	10^2	10^3	10^4	10^5
Czas [sek]	0,000000950	0,00001407	0,00017691	0,00297308	0,0424881

Zestawienie sortowania szybkiego z sortowaniem przez scalanie - przypadek średni jest przedstawiony na Rysunku 4.



Rysunek 4: Zestawienie sortowania szybkiego z sortowaniem przez scalanie - przypadek średni.

5 Wnioski:

To zadanie pokazuje, jak trudna jest praca na kodzie innej osoby oraz uświadamia początkującego programistę, jak ważna w postaci całego programu, jest dobrze wykonana dokumentacja.

Implementacja sortowania szybkiego na liście opartej na węzłach może być kosztowna ze względu na brak możliwości efektywnego indeksowania elementów. Jednakże złożoność obliczeniowa sortowania szybkiego na liście dwukierunkowej opartej na węzłach jest taka sama jak w przypadku sortowania listy zaimplementowanej na tablicy - $O(n \log n)$ dla przypadku średniego oraz $O(n^2)$ dla przypadku pesymistycznego. Przypadek pesymistyczny występuje w momencie, gdy chcemy posortować listę już posortowaną. W odróżnieniu do listy opartej o tablicę, w tej implementacji nie można w efektywny sposób dobrać pivotu w losowy sposób. Sortowanie szybkie na liście na węzłach może zostać przeprowadzone tylko gdy mamy pivotu ustawionego na stałe (np. jako ostatni element listy).

Złożoność obliczeniowa sortowania przez scalanie dla przypadku średniego jest taka sama jak dla przypadku pesymistycznego - $O(n \log n)$, czyli taka sama jak dla przypadku średniego przy sortowaniu szybkim.

Na wykresie (Rysunek 4.) widać, że algorytm quicksort jest niewiele szybszy niż mergesort, aczkolwiek dzięki temu, że sortowanie szybkie jest algorytmem 'in-place' (a mergesort - nie), czyli w przeciwieństwie do sortowania przez scalanie, nie wykorzystuje dodatkowej przestrzeni pamięci, jest najczęściej wykorzystywanym algorytmem do sortowania.