Laboratorium Projektowania algorytmów i metod sztucznej inteligencji

Quicksort, mergesort

218582

11 kwietnia 2016

Spis treści

Spis rysunków

Spis tablic

1 Wstęp

Celem zadania było:

- Implementacja algorytmu sortowania szybkiego i analiza złożoności $O(\cdot)$ dla przypadku średniego i pesymistycznego;
- Sprawdzenie 2-3 metod wybierania piwota w algorytmie quicksort i ich wpływ na $O(\cdot)$;
- Implementacja sortowania przez scalanie i analiza złożoności $O(\cdot)$.

2 Quicksort

Streszczenie

W tej części zostały omówione użyte rozwiązania oraz wyniki badań złożoności dla algorytmu sortowania szybkiego.

2.1 Przypadek średni a przypadek pesymistyczny

Algorytm sortowania szybkiego w średnim przypadku charakteryzuje się złożonością $O(nlogn)$, podczas gdy dla przypadku pesymistycznego $O(n^2)$. Jest to znaczna różnica, szczególnie dla dużej ilości elementów.

Przypadkiem pesymistycznym jest ten, w którym każde kolejne wywołanie rekurencyjne spowoduje wybór najmniejszego lub największego dostępnego elementu za piwot. Wtedy trzeba wykonać najwięcej operacji, nawet gdy tablica jest już posortowana.

Wymuszenie przypadku pesymistycznego następuje poprzez próbę posortowania tablicy już posortowanej i wybierania za każdym razem najmniejszej lub największej wartości.

Przypadek średni to taki, w którym wykorzystuje się standardowe metody, omówione w sekcji 2.2 *Wybrane metody piwotowania*, dla losowych danych. Zdecydowano, że danymi do sortowania będą pseudolosowe wartości typu integer od 0 do 9999.

2.2 Wybrane metody piwotowania

Zdecydowano zbadać następujące metody piwotowania:

- elementem środkowym w danej, nieposortowanej tablicy,
- elementem losowym,
- elementem będącym medianą wartości: elementu pierwszego, ostatniego i wybranego losowo z nieposortowanej tablicy.

2.3 Wyniki

Podsumowując, wykonano następujące badania:

- Przypadek średni - wybór losowego elementu za piwot z tablicy losowych wartości (sekcja 2.3.1),
- Przypadek pesymistyczny - wybór ostatniego elementu za piwot z tablicy posortowanych wartości (sekcja 2.3.1),
- Różne metody wyboru piwota: porównanie (sekcja 2.3.2)

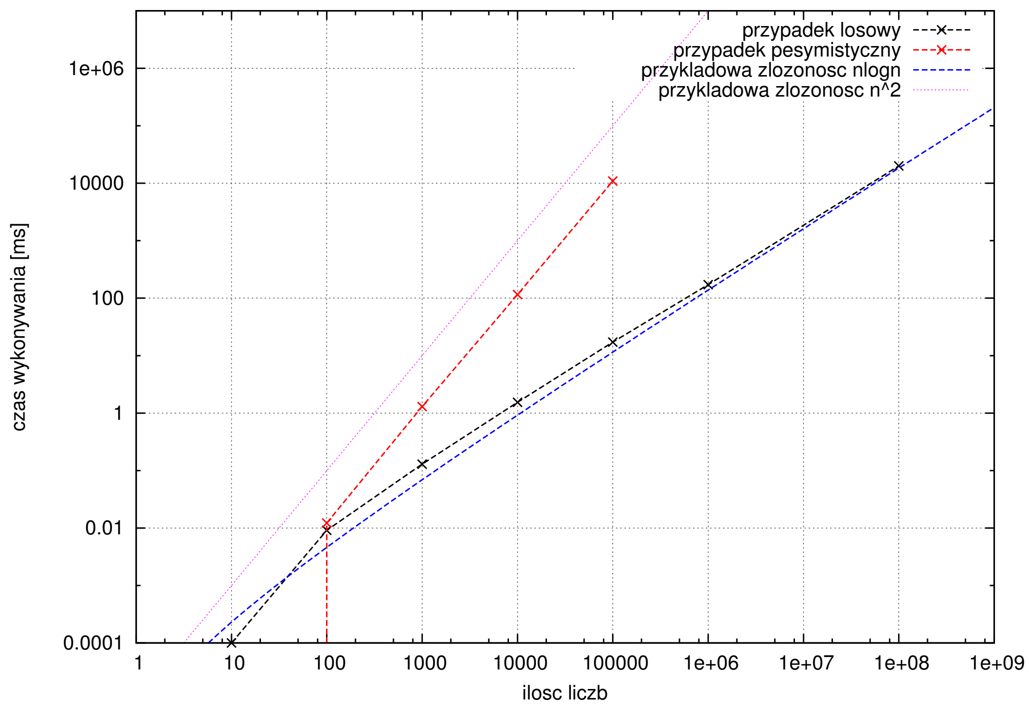
– piwot - losowy element,
– piwot - środkowy element,
– piwot - element, którego wartość jest medianą trzech wartości: watrości elementu pierwszego, ostatniego i losowego.

2.3.1 Przypadek średni i pesymistyczny

Porównano czasy działania dla przypadku średniego i pesymistycznego. Wyniki prezentuje tablica 1 oraz rysunek 1.

Tablica 1: Porównanie - przypadek średni i pesymistyczny

Ilość	Czas [ms]	
	Średni	Pesymistyczny
10^1	0.0001	0.0001
10^2	0.0091	0.0122
10^3	0.1291	1.2966
10^4	1.5464	115.939
10^5	17.1818	10886.4
10^6	171.67	-
10^8	19988.1	-



Rysunek 1: Porównanie przypadków - pesymistycznego i średniego

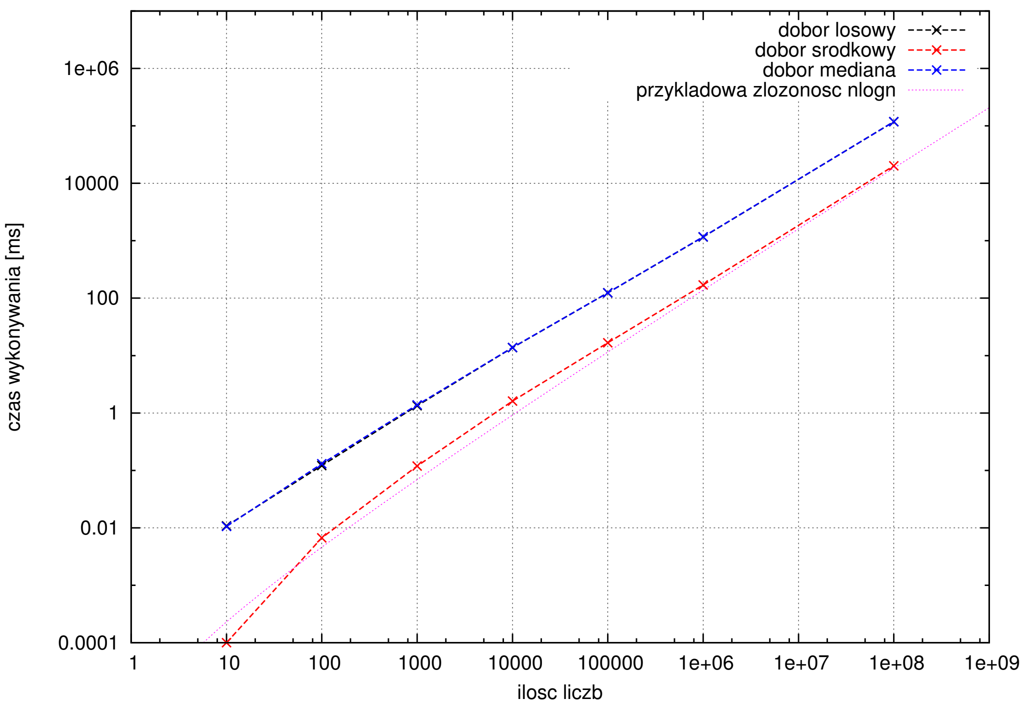
Można odczytać z wykresu, że przypadek pesymistyczny charakteryzuje się złożonością $O(n^2)$, podczasy gdy średni - $O(nlogn)$. Różnica ta jest znacząca, jeśli chodzi o czas działania algorytmu.

2.3.2 Porównanie metod piwotowania

W tablicy 2 znajdują się uśrednione wyniki pomiarów dla kolejnych wielkości problemu, dla różnych sposobów wyboru piwota. Środkowy - oznacza wybór elementu znajdującego się w połowie danego fragmentu tablicy za piwot; losowy - wybór losowego elementu za piwot; mediana - wybór elementu o wartości równej medianie wartości trzech elementów - pierwszego, losowego i ostatniego. Wykres znajdujący się na rysunku 2 prezentuje dane zawarte w wyżej wymienionej tablicy, a także wykres przykładowej złożoności $O(nlogn)$ dla odniesienia.

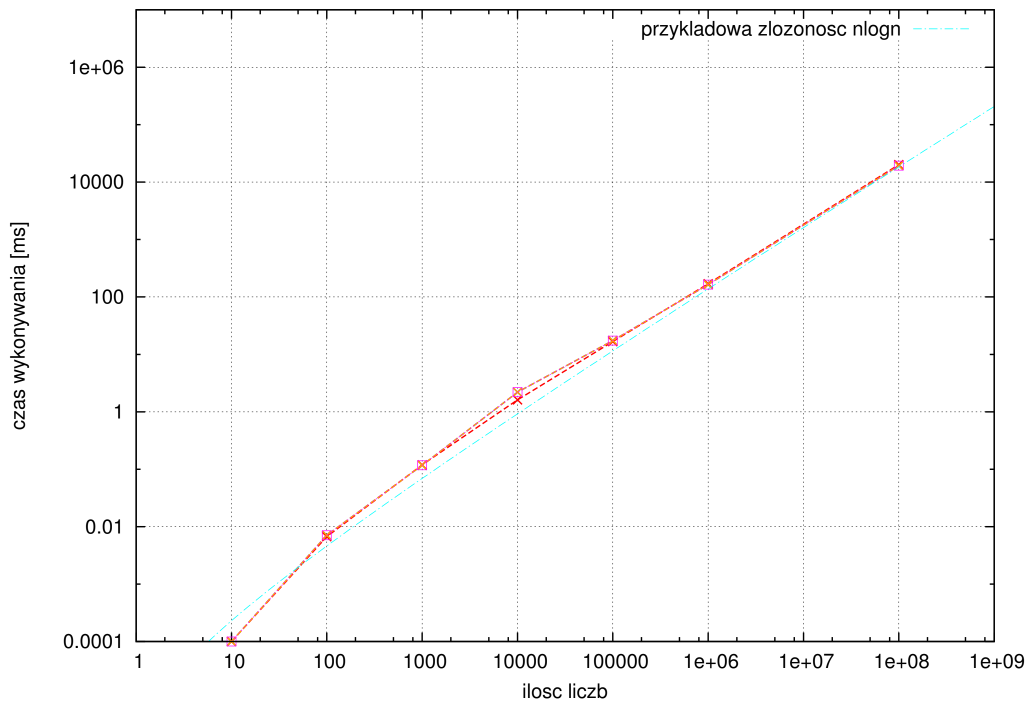
Tablica 2: Różne metody piwotowania

Ilość	Czas [ms]		
	Środkowy	Losowy	Mediana
10^1	$< 1\mu s$	0.0108	0.0106
10^2	0.0067	0.1213	0.1309
10^3	0.1191	1.3395	1.3865
10^4	1.618	13.7723	13.8558
10^5	16.6846	123.656	123.925
10^6	169.673	1162.37	1169.73
10^7	20179.5	118056	118077



Rysunek 2: Czas działania kilku metod piwotowania dla kolejnych wielkości problemów. Dobór losowy i dobór mediana niemal się pokrywają

Widać zatem wyraźnie, że wszystkie trzy sposoby charakteryzują się złożonością $O(nlogn)$ dla przypadku średniego. Wybór środkowego elementu z tablicy okazuje się być najszybszy. Zauważyć tutaj można jedynie niewielkie odchylenie od teoretycznego przebiegu dla $O(nlogn)$ w środkowej części badanego zakresu. Powtórzenie kilkukrotne tego badania pozwoliło uzyskać podobne rezultaty (Rys. 3)



Rysunek 3: Powtórzenie obliczeń dla wyboru środkowego elementu jako piwota

3 Mergesort

Streszczenie

W tej części zostały przedstawione wyniki badań złożoności dla algorytmu sortowania przez scalanie.

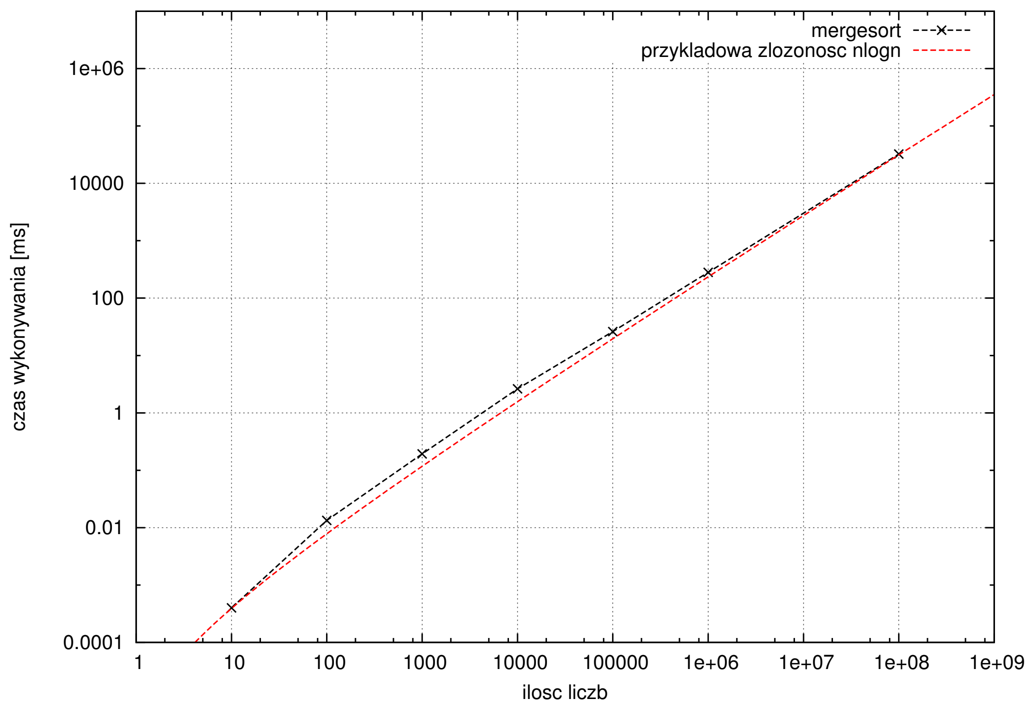
Algorytm sortowania przez scalanie charakteryzuje się złożonością $O(nlogn)$. W tej części przeprowadzono badania mające na celu potwierdzenie złożoności napisanego algorytmu.

Wykonano dziesięciokrotne powtórzenie pomiaru dla każdej zmierzonej ilości sortowanych liczb i wynik uśredniono.

Tablica 3 oraz rysunek 4 prezentują wyniki wykonanych badań.

Tablica 3: Czas działania algorytmu mergesort dla różnych ilości danych

Ilość	Czas [ms]
10^1	0.0004
10^2	0.0134
10^3	0.1925
10^4	2.6264
10^5	26.0681
10^6	281.341
10^8	32449.1



Rysunek 4: Czas działania algorytmu mergesort

4 Wnioski

4.1 Quicksort

- Przypadek pesymistyczny zmienia złożoność algorytmu sortowania szybkiego do $O(n^2)$ ze średniego $O(nlogn)$. Tak znacząca zmiana wymaga podjęcia kroków celem zapobieżenia natrafienia na przypadek pesymistyczny.
- Sposób doboru piwota ma główny wpływ na czas działania algorytmu. Przede wszystkim należy stosować taki dobór piwota, który znacząco zmniejszy prawodpodobieństwo natrafienia na przypadek pesymistyczny.
- Dobór losowy piwota i dobór według mediany elementów pierwszego, ostatniego i losowego niemal nie różnią się jeśli chodzi o czasu wykonywania.
- Dobór środkowego elementu wydaje się być najszybszy ze wszystkich. Prawdopodobnym powodem jest fakt, że nie trzeba wykonywać losowania w celu wyboru piwota.
- Dla średniej ilości elementów (10^2 - 10^4) można zauważyć nieznaczne odchylenie od teoretycznego nachylenia wykresu dla złożoności $O(nlogn)$. Prawodpodobnie wynika to z faktu, że dla małych zbiorów istnieje większe prawdopodobieństwo natrafienia na przypadek bliski pesymistycznemu.

4.2 Mergesort

- Zaimplementowany algorytm mergesort charakteryzuje się złożonością $O(nlogn)$.