

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 2 – Algorytmy sortowania

Bartosz Rudyk 241160

1. Zadanie do wykonania

Celem zadania było napisanie 3 algorytmów sortowania (w tym dwóch o złożoności nie większej niż $O(n \log n)$) oraz przetestowanie ich wydajności dla różnych rozmiarów problemów.

2. Przygotowanie

Przygotowane algorytmy sortowania:

- Bouble Sort

Polega na porównaniu ze sobą 2 elementów i zamianie ich miejscami jeżeli ich kolejność względem siebie nie zgadza się. Algorytm kończy sortowanie jeżeli po przejściu przez całą tablicę nie znajdzie żadnego elementu, którego kolejność względem sąsiedniego nie zgadza się. Złożoność czasowa jest równa $O(n^2)$.

- Quick Sort

Z tablicy wybierany jest jeden element, który dzieli tablicę na dwie mniejsze. Następnie elementy z lewej tablicy, które są większe od elementu dzielącego są przetrzucane na jego prawą stronę. Analogiczne działanie jest podejmowane względem prawej tablicy, mniejsze elementy są przetrzucane na lewą stronę elementu odniesienia. Kolejnym działaniem jest rekurencyjne wywołanie funkcji dla lewej i prawej tablicy. Złożoność czasowa jest równa $O(n \log n)$.

- Merge Sort

W pierwszej kolejności, algorytm merge sort dzieli tablicę na dwie równe części, dopóki każda część nie będzie się składała z jednego elementu. Następnie części są ze sobą scalane w kolejności opartej o porównanie ich do siebie i sprawdzenie, która jest większa. Złożoność czasowa jest równa $O(n \log n)$.

3. Opis testów

Każdy algorytm został przetestowany na tych samych tablicach. Rozmiary tablic zmieniały się z każdym obrotem pętli o 1000 elementów, zakres rozmiarów tablic był od 1000 do 50000.

Dla każdego rozmiaru tablicy 10 razy były losowane różne wartości tablicy, za pomocą funkcji `rand()` i każdy algorytm sortował każdą tablicę po jednym razie. Miało to na celu wyliczenie średnich czasów sortowania.

Bouuble Sort

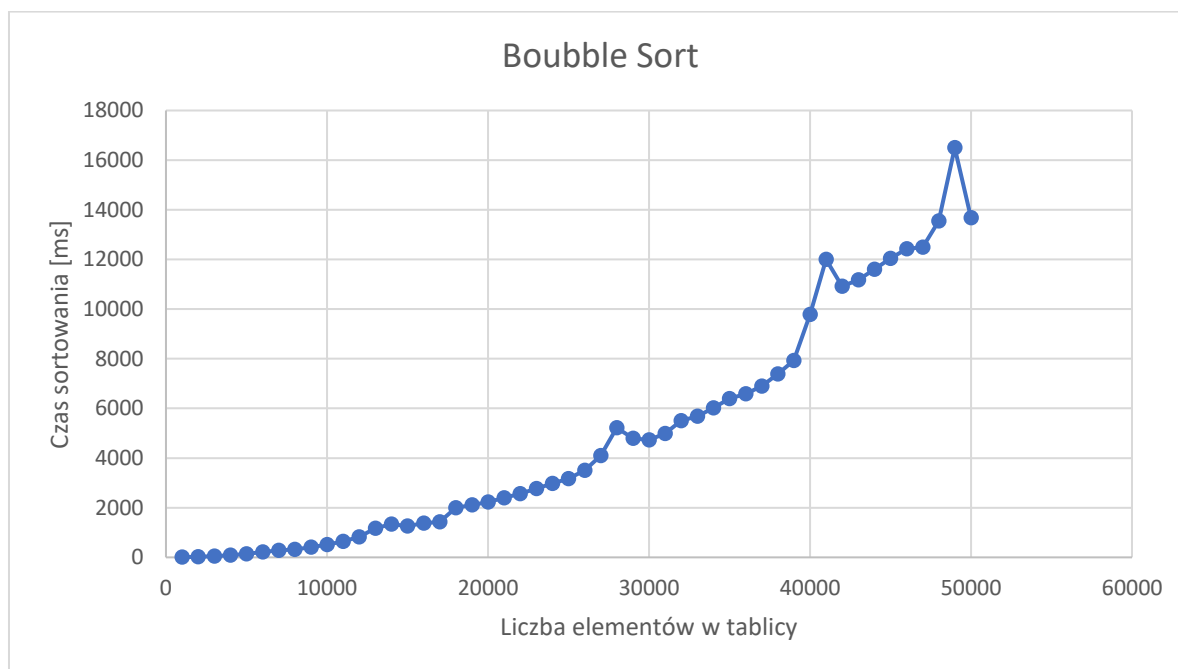


Tabela z początkowymi wartościami:

Czas[ms]	n
6.007687	1000
21.20146	2000
46.90471	3000
92.56583	4000
135.5658	5000
221.4655	6000
279.6838	7000
314.7031	8000
402.9559	9000
507.3735	10000

Quick Sort

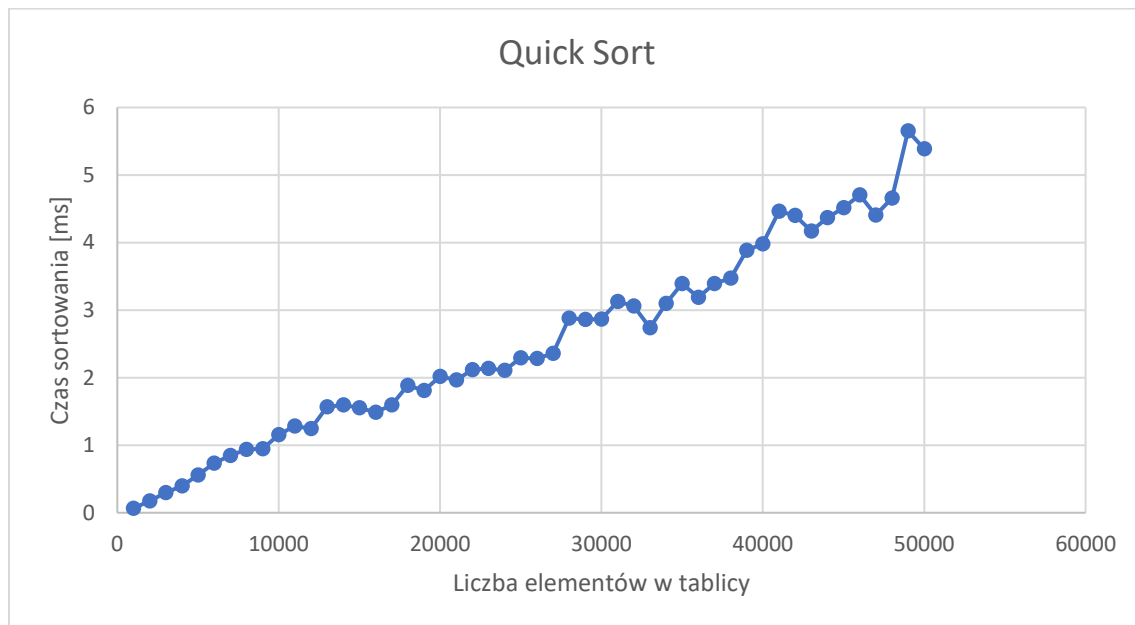


Tabela z początkowymi wartościami:

Czas[ms]	n
0.065333	1000
0.173022	2000
0.296134	3000
0.395511	4000
0.558445	5000
0.7352	6000
0.8452	7000
0.937333	8000
0.948889	9000
1.156488	10000

Merge Sort

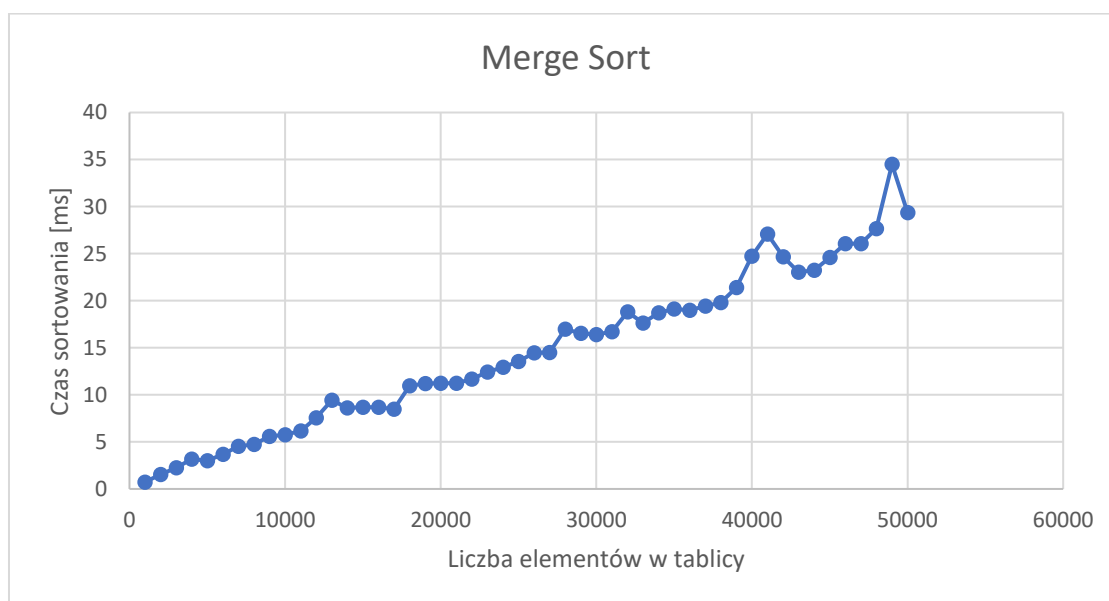


Tabela z początkowymi wartościami:

Czas[ms]	n
0.737333	1000
1.553468	2000
2.259822	3000
3.167333	4000
3.002222	5000
3.680133	6000
4.516445	7000
4.722	8000
5.578489	9000
5.769511	10000

4. Wnioski

Na podstawie wyników badań, dla każdego algorytmu został sporządzony wykres zależności czasu sortowania od ilości elementów w tablicy. Analizując krzywe wykresów można wyciągnąć wnioski, że złożoność czasowa algorytmów badanych w ramach ćwiczenia wykazuje podobieństwo do złożoności czasowej przyjętej w założeniach.

Na podstawie średnich czasów wykonania sortowań można stwierdzić, że quick sort jest najszybciej sortującym algorytmem. Kolejnym wnioskiem, który się nasuwa jest fakt jak bardzo nieefektywny pod względem czasu sortowania jest bubble sort, którego czas sortowania znacznie przewyższa czasy pozostałych algorytmów.