

Projektowanie algorytmów i metody sztucznej inteligencji

Projekt 2 – Algorytmy sortowania

8.04.2021, godz. 15:15

Aleksandra Rozmus 252954

Spis treści

1	Pomiary	2
1.1	Dla złożoności obliczeniowej $O(n \log n)$	2
1.2	Dla złożoności obliczeniowej $O(n^2)$	3
2	Wygenerowane wykresy	4
2.1	Sortowanie szybkie	4
2.2	Pesymistyczny przypadek sortowania szybkiego	4
2.3	Sortowanie przez scalanie	5
2.4	Sortowanie bąbelkowe	6
3	Wnioski	6

1 Pomiary

1.1 Dla złożoności obliczeniowej $O(n \log n)$

Tablica 1: Średni czas sortowania w zależności od ilości elementów dla algorytmów o złożoności obliczeniowej $O(n \log n)$

	Sortowanie szybkie	Sortowanie przez scalanie
ilość elementów [n]	średni czas sortowania [s]	średni czas sortowania [s]
50000	0,008467	0,030295
100000	0,017867	0,0624
150000	0,026874	0,094694
200000	0,036612	0,128652
250000	0,045976	0,16125
300000	0,055718	0,195545
350000	0,065357	0,22835
400000	0,074942	0,263332
450000	0,08478	0,297684
500000	0,094323	0,329404
550000	0,103894	0,361053
600000	0,113568	0,399798
650000	0,124131	0,430567
700000	0,132877	0,465933
750000	0,14495	0,503174
800000	0,154476	0,536354
850000	0,164126	0,568357
900000	0,172471	0,605494
950000	0,184735	0,642559
1000000	0,194463	0,676947

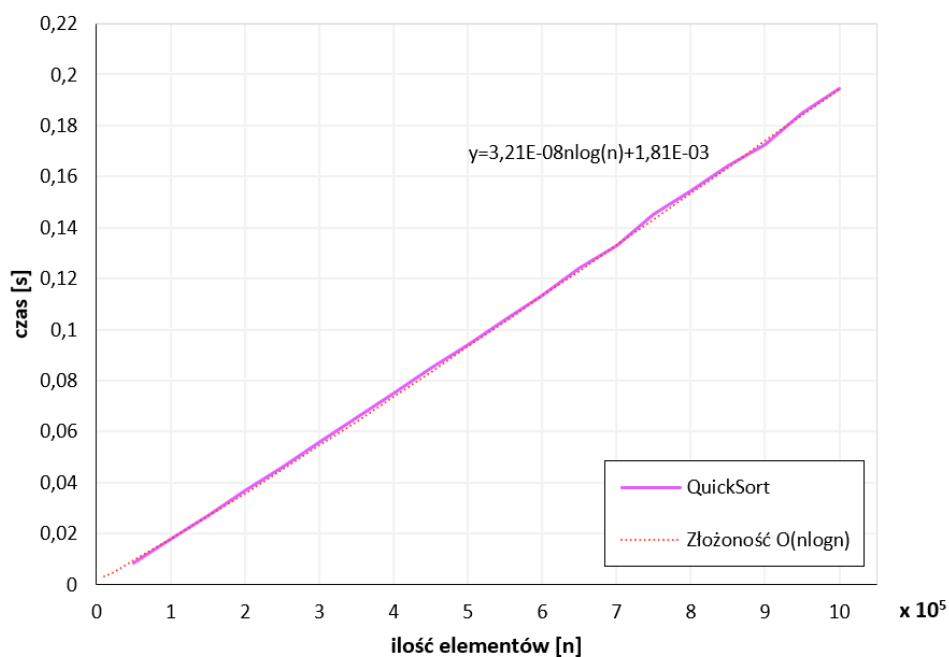
1.2 Dla złożoności obliczeniowej $O(n^2)$

Tablica 2: Średni czas sortowania w zależności od ilości elementów dla algorytmów o złożoności obliczeniowej $O(n^2)$

	Pesymistyczne sortowanie szybkie	Sortowanie bąbelkowe
ilość elementów [n]	średni czas sortowania [s]	średni czas sortowania [s]
5000	0,024931	0,18969
10000	0,095063	0,757415
15000	0,204338	1,70636
20000	0,346914	3,030724
25000	0,523276	4,728682
30000	0,716602	6,813054
35000	0,93538	9,262389
40000	1,169786	12,08112
45000	1,424951	15,2907
50000	1,699823	18,88341
55000	1,966504	22,67286
60000	2,275482	27,17305
65000	2,574706	31,88604
70000	2,895536	36,9804
75000	3,228837	42,4627

2 Wygenerowane wykresy

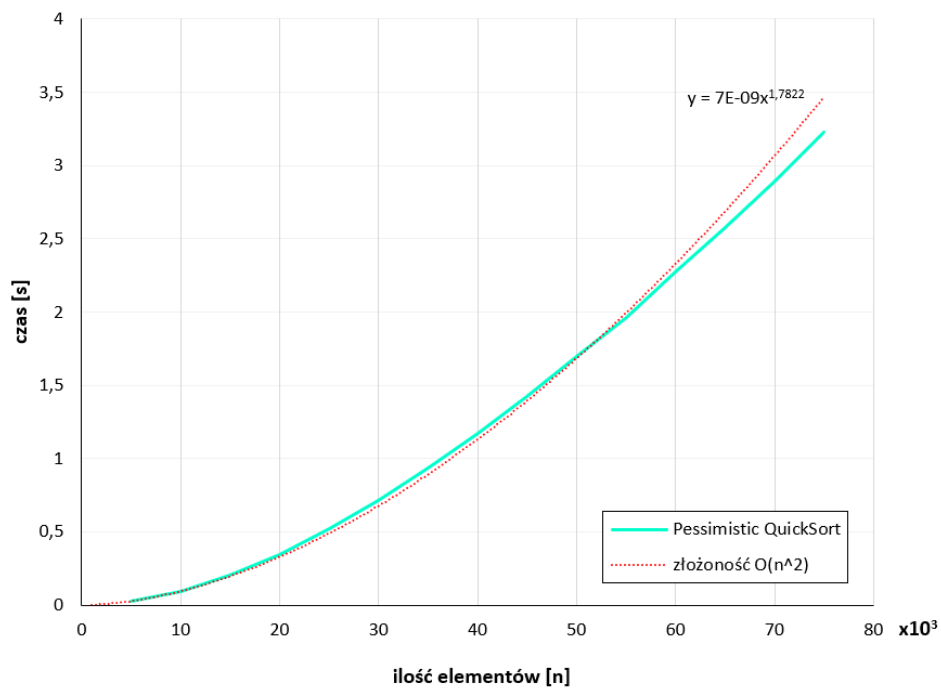
2.1 Sortowanie szybkie



Rysunek 1: Wykres czasu wykonania algorytmu sortowania szybkiego w zależności od liczby danych do posortowania

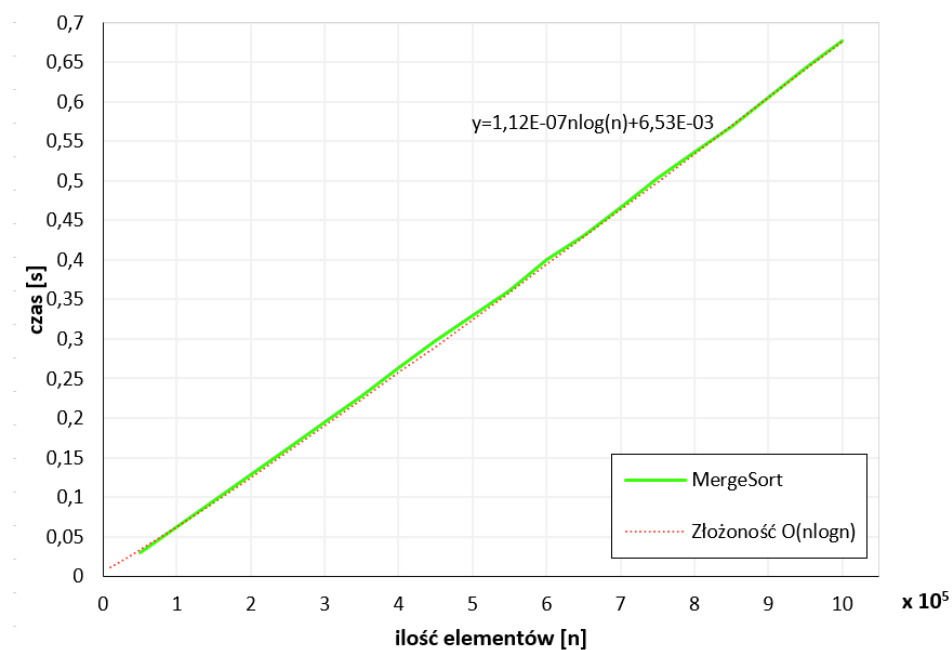
2.2 Pesymistyczny przypadek sortowania szybkiego

Istnieje pesymistyczny przypadek sortowania szybkiego, dla którego przyjmuje ono złożoność obliczeniową $O(n^2)$. Dzieje się to, kiedy jako pivot wybierzemy pierwszy, największy element z podanej tablicy, która wcześniej została posortowana w kolejności od największego do najmniejszego elementu.



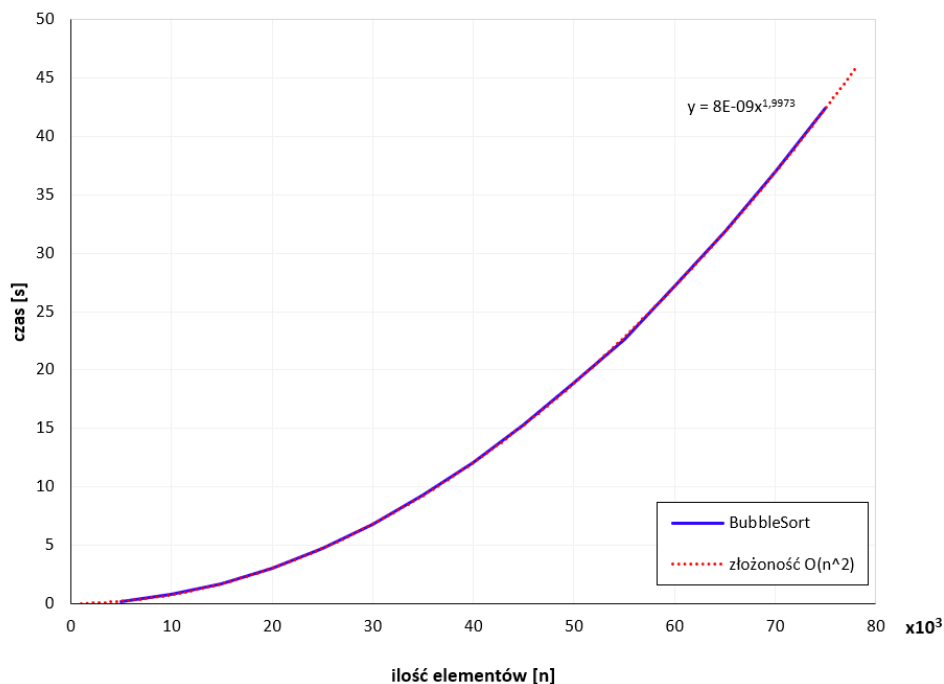
Rysunek 2: Wykres czasu wykonania algorytmu sortowania szybkiego w zależności od liczby danych do posortowania dla pesymistycznego przypadku

2.3 Sortowanie przez scalanie



Rysunek 3: Wykres czasu wykonania algorytmu sortowania przez scalanie w zależności od liczby danych do posortowania

2.4 Sortowanie bąbelkowe



Rysunek 4: Wykres czasu wykonania algorytmu sortowania bąbelkowego w zależności od liczby danych do posortowania

3 Wnioski

- Algorytmy o złożoności obliczeniowej $O(n \log n)$ są zdecydowanie szybsze od algorytmów o złożoności $O(n^2)$.
- Algorytm sortowania szybkiego o złożoności $O(n \log n)$ okazał się najszybszy z wybranych algorytmów.
- Algorytm sortowania bąbelkowego o złożoności $O(n^2)$ okazał się najwolniejszy z wybranych algorytmów.
- W zależności od sposobu wyboru pivotu oraz niekorzystnego ułożenia danych wejściowych złożoność obliczeniowa algorytmu sortowania szybkiego degraduje się do $O(n^2)$. Należy również uważać, ponieważ może występować przepełnienie stosu (w moim przypadku nastąpiło to przy próbie wywołania 25 powtórzeń sortowania dla zakresu $< 5000, 100000 >$ elementów z krokiem co 5 000).
- Wygenerowane przebiegi czasowe wykonania algorytmów zgadzają się z założonymi przebiegami złożoności obliczeniowej.