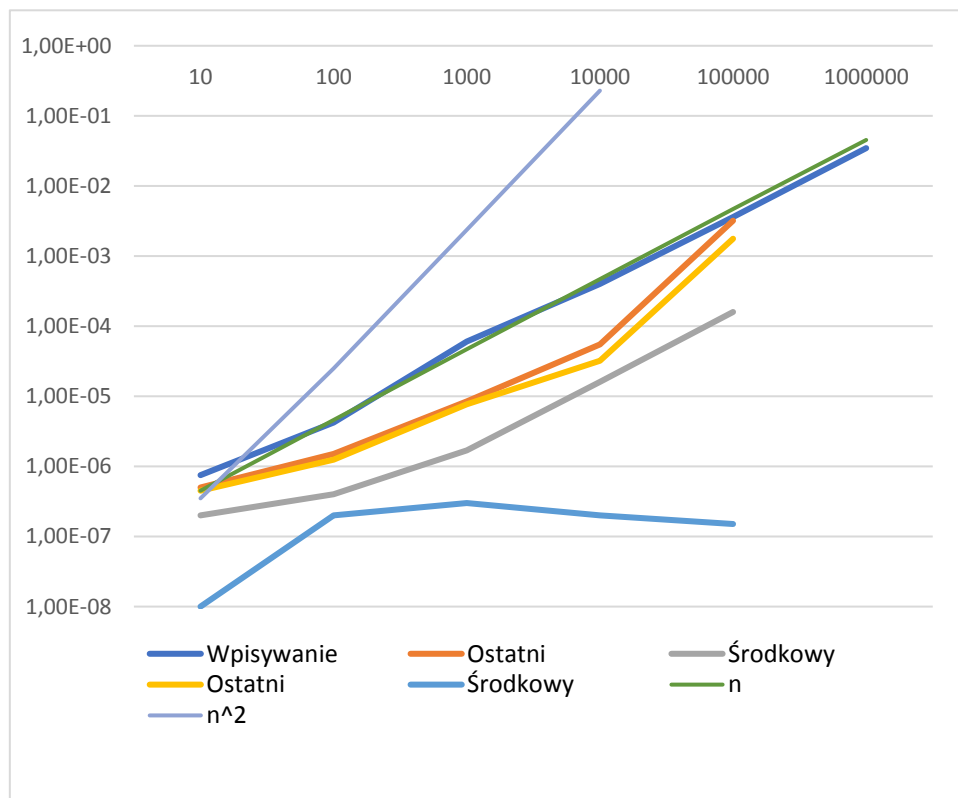


BST

Ćwiczenie polegało na zaimplementowaniu wybranego drzewa i zbadanie jego złożoności obliczeniowych.

Dla BST teoretyczne złożoności obliczeniowe podstawowych operacji to:
 dodawanie elementu $O(n)$, znajdowanie elementu $O(n)$ – pesymistyczny,
 $O(\log n)$ – średni

n	Wpisywanie	Wyszukiwanie				Alogorytmy o złożoności	
		Rosnące dane		Malejące dane		n	n ²
		Ostatni	Środkowy	Ostatni	Środkowy		
10	7,50E-07	5,00E-07	2,00E-07	4,50E-07	1,00E-08	4,50E-07	3,50E-07
100	4,20E-06	1,50E-06	4,00E-07	1,25E-06	2,00E-07	4,60E-06	2,50E-05
1000	6,01E-05	8,45E-06	1,70E-06	7,70E-06	3,00E-07	4,70E-05	2,37E-03
10000	4,01E-04	5,50E-05	1,60E-05	3,20E-05	2,00E-07	4,68E-04	2,29E-01
100000	3,60E-03	3,20E-03	1,59E-04	1,77E-03	1,50E-07	4,68E-03	
1000000	3,46E-02					4,53E-02	



Wnioski

Z wykresu wynika, że złożoność obliczeniowa badanej implementacji drzewa dla wpisywania jest równa $O(n)$. Widać także, że złożoność wyszukiwania jest mniejsza niż $O(n)$, można więc wnioskować, że złożoność ta to $O(\log n)$.

Pomiary są zgodne z założeniami teoretycznymi, więc badana implementacja jest poprawna.

W badaniach nie udało się uzyskać najgorszej możliwej złożoności wyszukiwania. Jest to przypadek gdy drzewo staje się listą, wtedy złożoność obliczeniowa wyszukiwania wynosiłaby $O(n)$. W binary search tree jest to możliwe, ponieważ nie stosuje się w nim bilansowania.