

Struktury Danych i Złożoność Obliczeniowa

Projekt 1

Sprawozdanie

Szymon Leja

Indeks: 259047

Zadanie nr 1	
Badanie efektywności operacji na danych w podstawowych strukturach danych	
Prowadzący kurs:	Dr inż. Zbigniew Buchalski
Termin zajęć:	TN Pon. 13:15
Termin oddania:	28.03.2022r.

1. Wstęp

Celem zadania było stworzenie programu, który będzie sprawdzał czas wykonywania operacji na różnych strukturach danych, m.in. dodawanie, usuwanie czy wyszukiwanie elementów.

Struktury użyte w projekcie:

- Tablica dynamiczna
- Lista dwukierunkowa
- Kopiec binarny (max)
- Drzewo BST

Zasady implementacji i testowania struktur:

- Program został stworzony przy użyciu C++, wszystkie struktury zostały przetestowane bez gotowych implementacji z zew. bibliotek
- Dane liczbowe były generowane przy użyciu generatora pseudolosowych liczb
- Struktury były wielokrotnie testowane, a wyniki czasowe zostały uśrednione
- Pomiar czasu został przygotowany przy użyciu biblioteki umożliwiającej sprawdzanie zegara w dużej rozdzielczości *chrono::high_resolution_clock*
- Struktury były testowane z pamięcią DDR4 16GB 3600MHz i procesorem Ryzen 5600X z taktowaniem 4.5GHz

2. Złożoność obliczeniowa

Złożoność obliczeniową jest określana jako ilość zasobów koniecznych do przetworzenia algorytmu przez komputer.

Wyróżnia się dwa typy złożoności obliczeniowej:

- Czasową – czas potrzebny na wykonanie algorytmu, jest uzależniona od wielkości danych na których operuje struktura.
- Pamięciową – ilość pamięci potrzebnej na wykonanie algorytmu bez której dany algorytm nie może zostać poprawnie wykonany, również jest uzależniona od wielkości danych wejściowych.

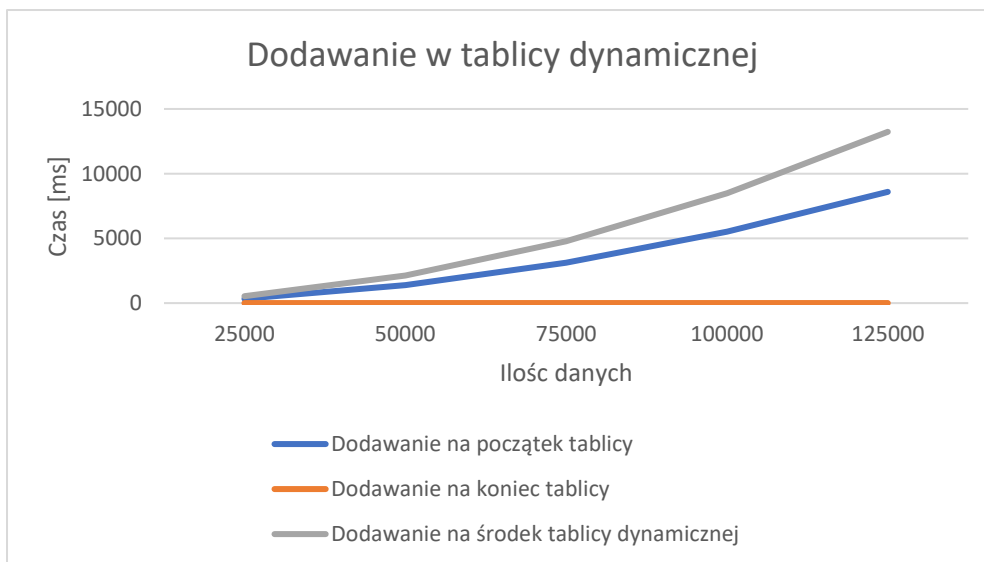
Ponadto wyróżnia się trzy „postrzegania” złożoności:

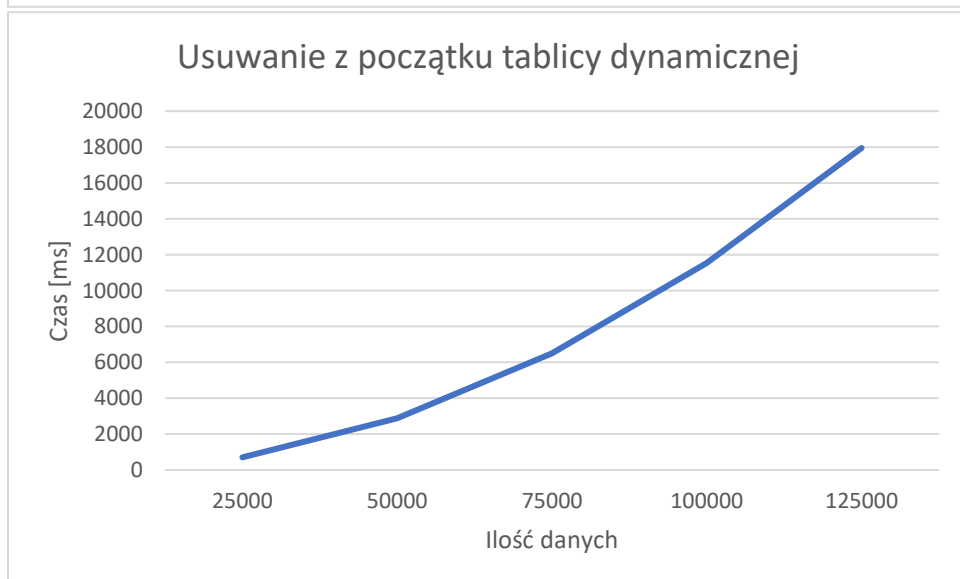
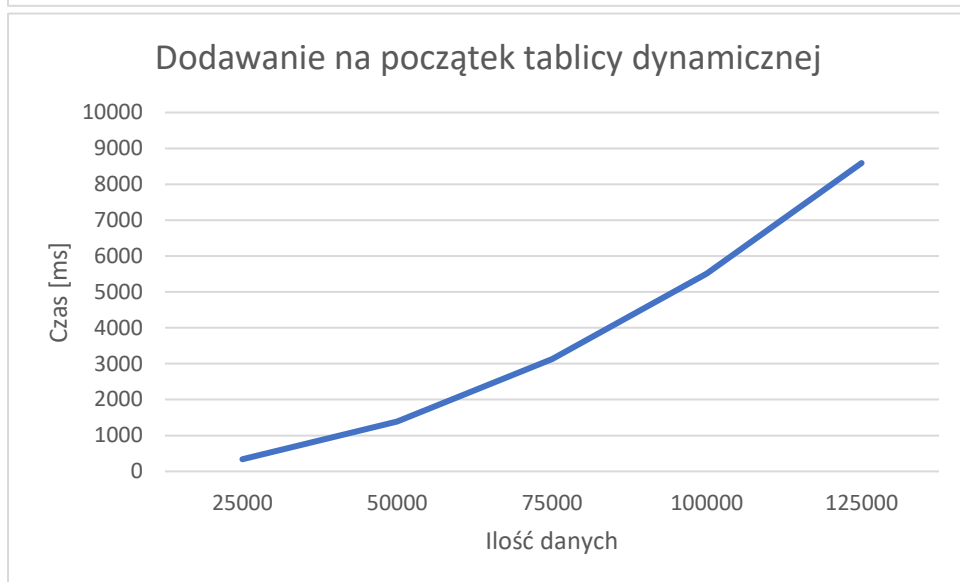
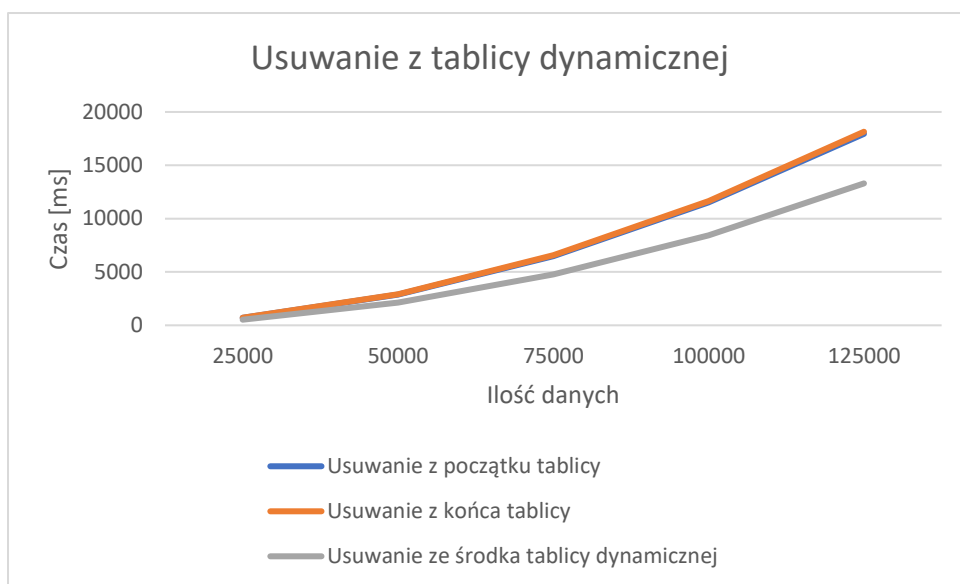
- Pesymistyczną – ilość czasu lub pamięci potrzebnej do wykonania algorytmu dla najmniej optymalnego zestawu danych.
- Średnią - ilość czasu lub pamięci potrzebnej do wykonania algorytmu dla zwykłego, losowego, typowego zestawu danych.
- Optymistyczną - ilość czasu lub pamięci potrzebnej do wykonania algorytmu dla najbardziej optymalnego zestawu danych.

Złożoność tablicy dynamicznej:

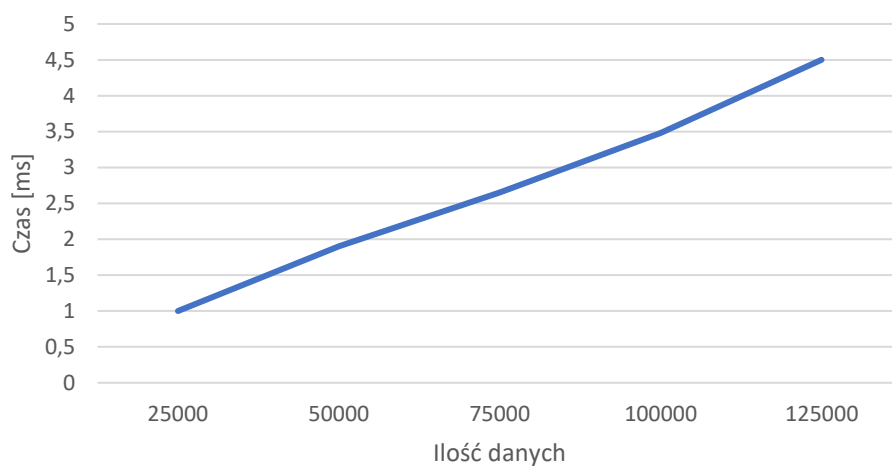
Funkcja	Średnia	Pesymistyczna
Dodanie	$O(n)$	$O(n)$
Usunięcie	$O(n)$	$O(n)$
Wyszukanie	$O(n)$	$O(n)$

Ilość danych	25000	50000	75000	100000	125000
Dodawanie (START) [ms]	338,047	1384,58	3132,12	5515	8593
Usuwanie (START) [ms]	704,09	2872,27	6495,21	11531,2	17947
Dodawanie (END) [ms]	1	1,9	2,65	3,48	4,5
Usuwanie (END) [ms]	711,3	2909	6557,52	11646	18145,4
Dodawanie (MID) [ms]	532,48	2119,94	4788,21	8475,78	13230
Usuwanie (MID) [ms]	527,48	2124,08	4775,86	8440,53	13300
Wyszukiwanie [ms]	205	729,946	1366,89	2066,82	2764,2

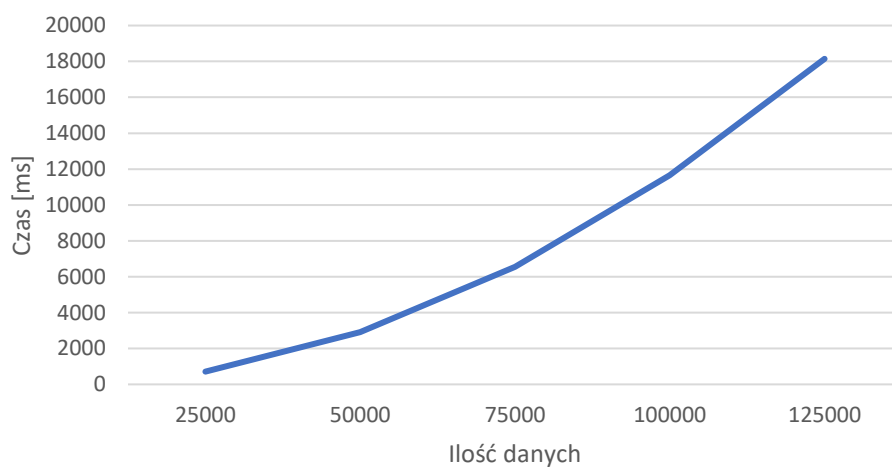




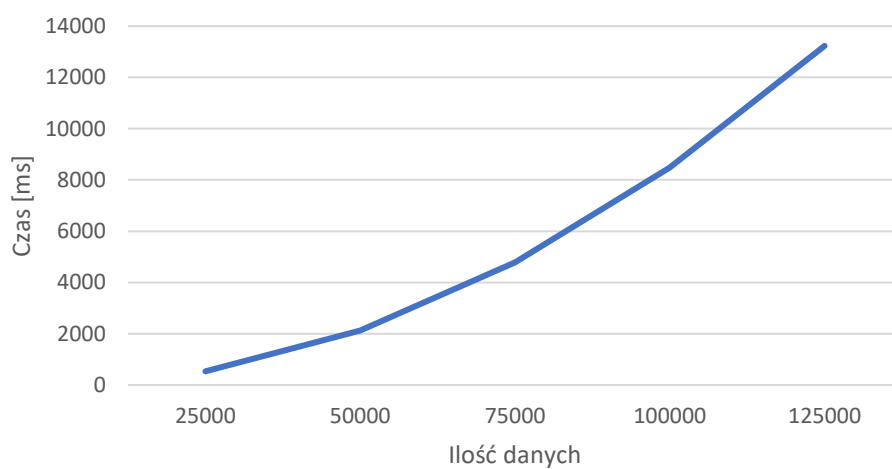
Dodawanie na koniec tablicy dynamicznej

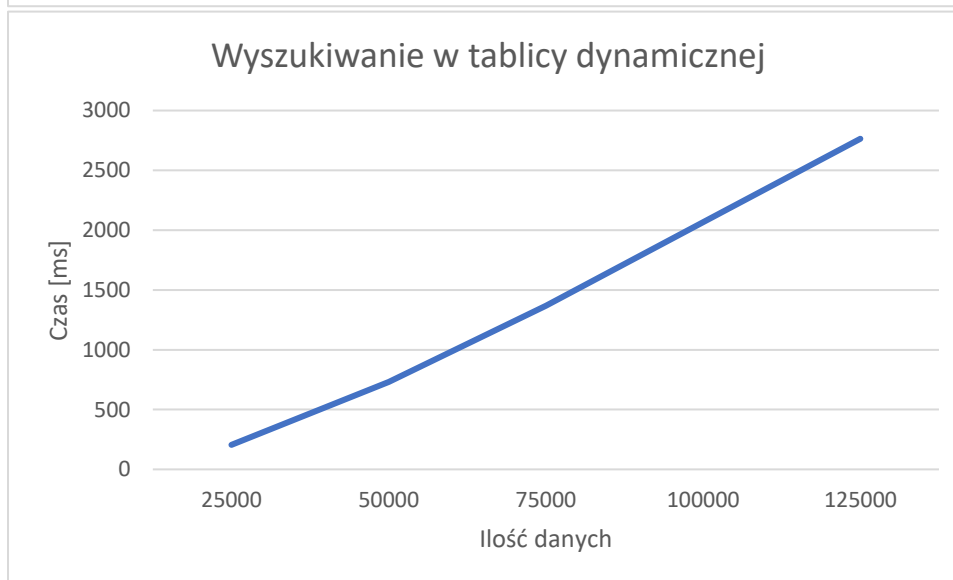
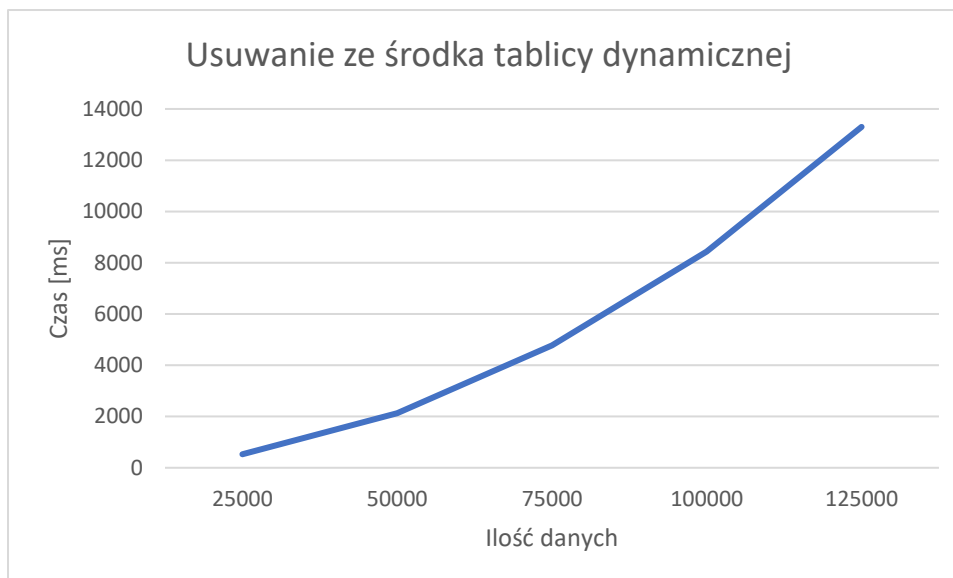


Usuwanie z końca tablicy dynamicznej



Dodawanie na środek tablicy dynamicznej



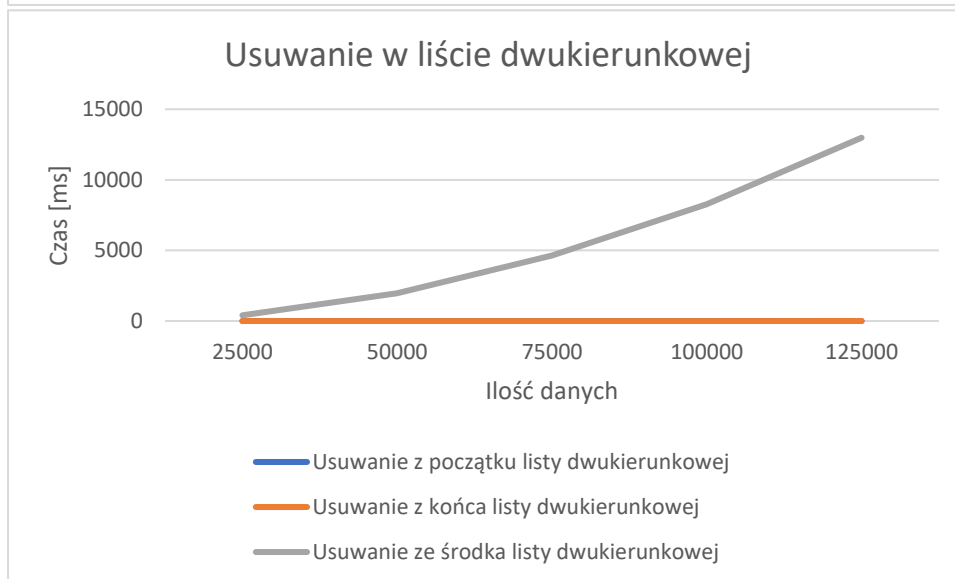
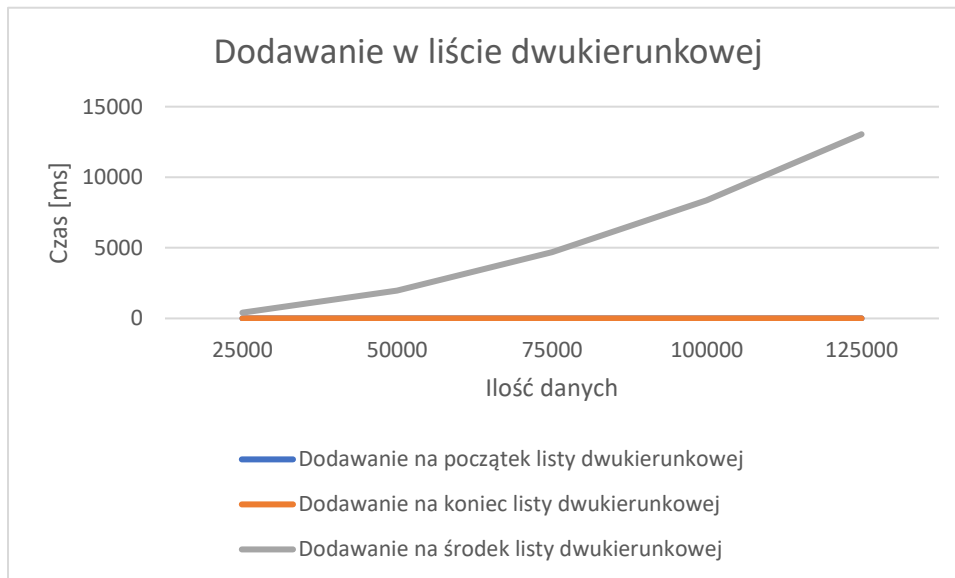


Złożoność listy dwukierunkowej:

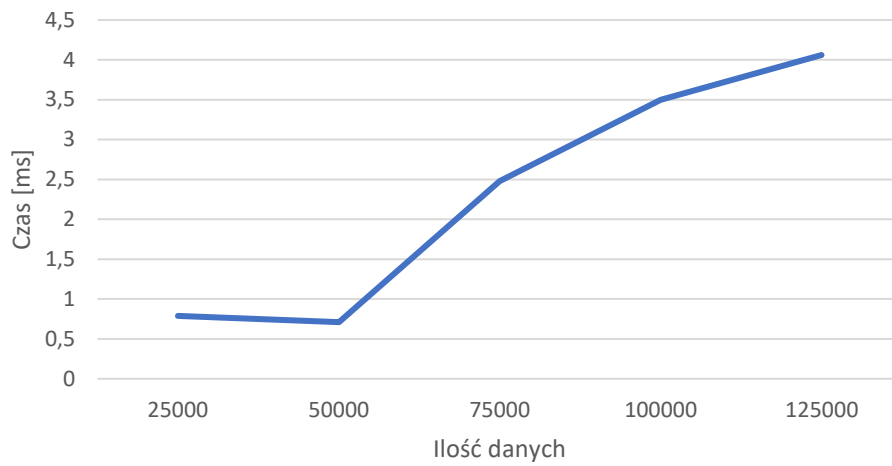
Funkcja	Średnia	Pesymistyczna
Dodanie	$O(-)$	$O(1)$
Usunięcie	$O(-)$	$O(1)$
Wyszukanie	$O(n)$	$O(n)$

Ilość danych	25000	50000	75000	100000	125000
Dodawanie (START) [ms]	0,79	0,71	2,48	3,5	4,06
Usuwanie (START) [ms]	0,45	0,9	1,5	1,89	2,41
Dodawanie (END) [ms]	0,83	1,7	2,43	3,25	3,99
Usuwanie (END) [ms]	0,33	0,9	1,47	1,75	2,66

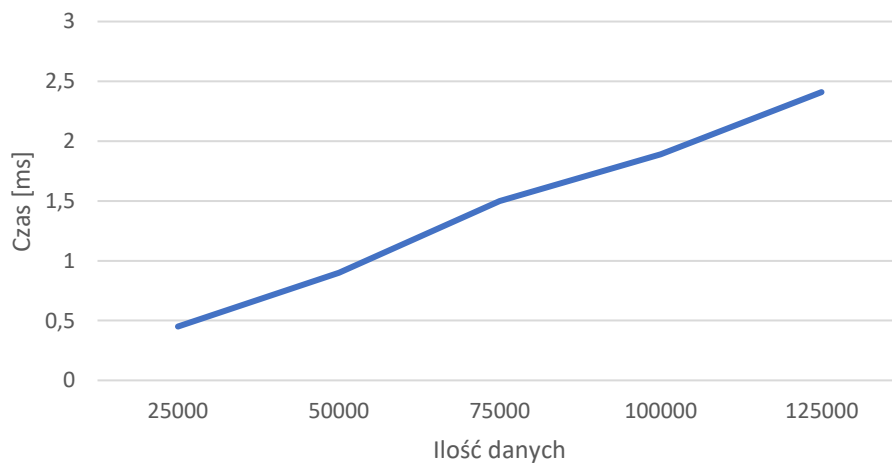
Dodawanie (MID) [ms]	409,36	1978,2	4697,4	8364,97	13045
Usuwanie (MID) [ms]	406,803	1967,9	4645,12	8271,15	12983,6
Wyszukiwanie [ms]	350.087	1327,71	2542,1	3777	5134,3



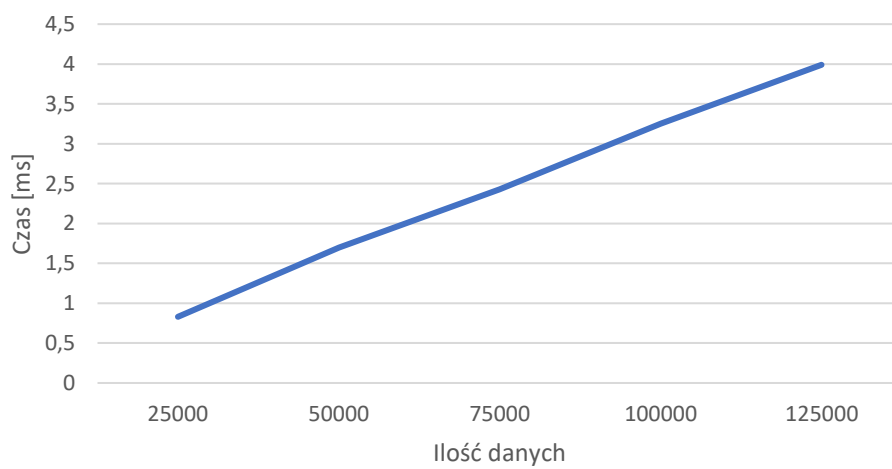
Dodawanie na początek listy dwukierunkowej



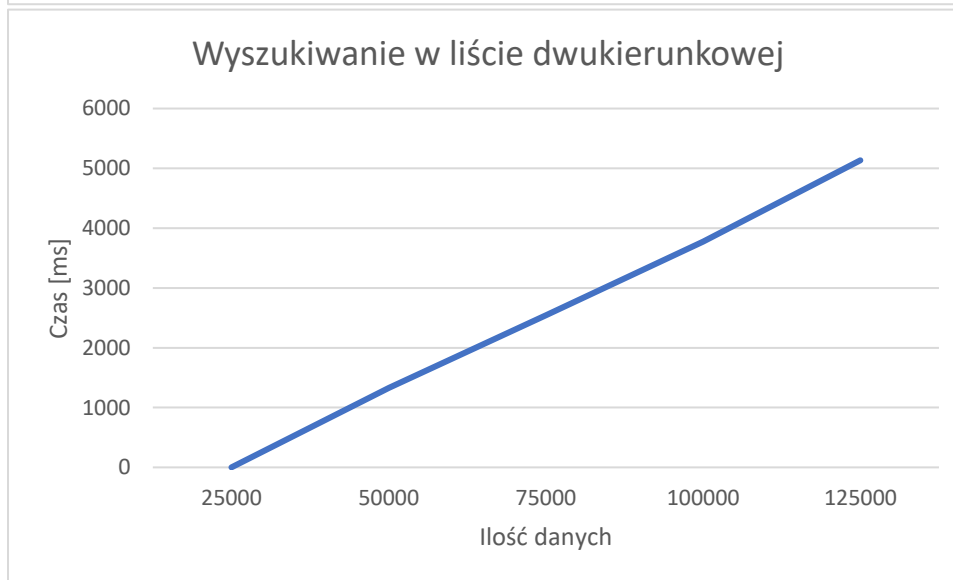
Usuwanie z początku listy dwukierunkowej



Dodawanie na koniec listy dwukierunkowej



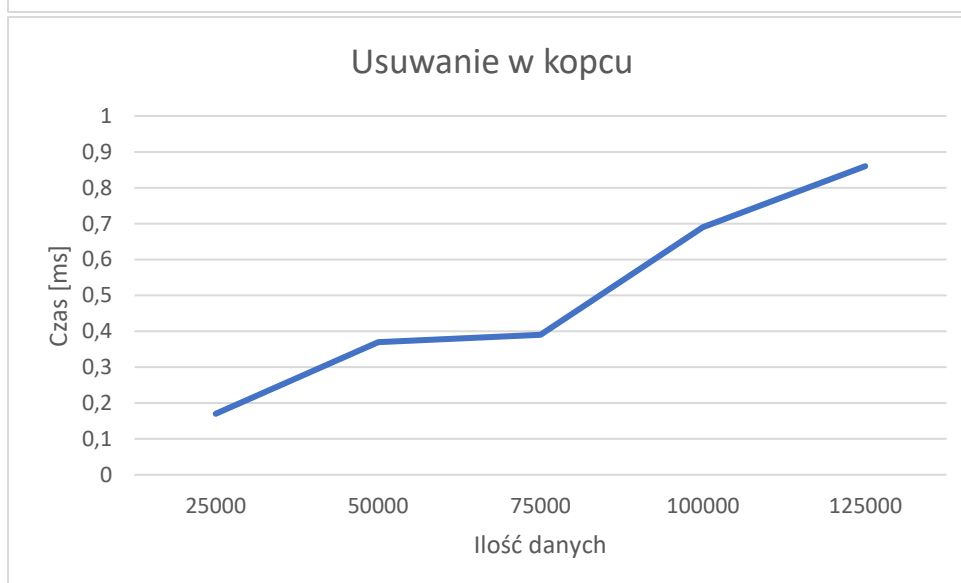
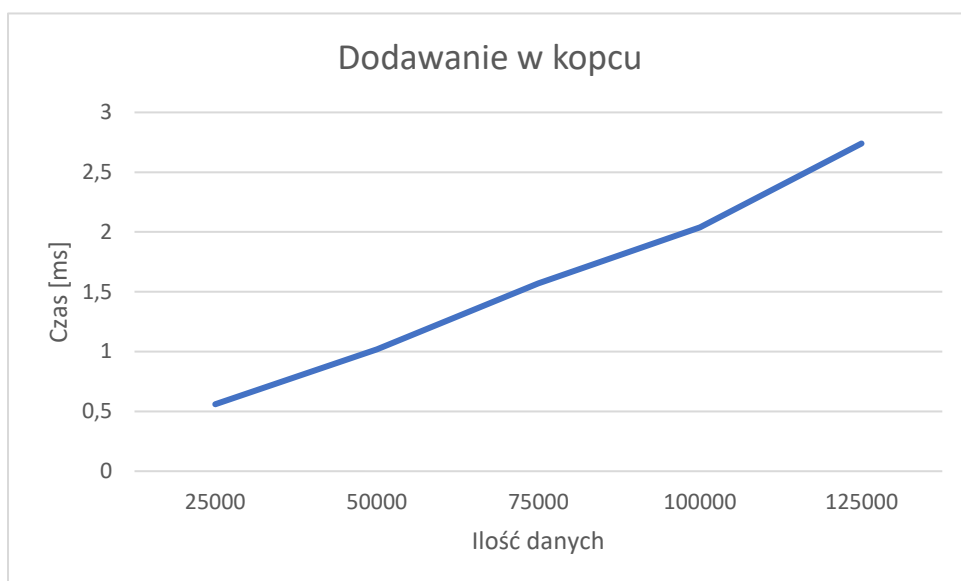




Złożoność kopca:

Funkcja	Średnia	Pesymistyczna
<u>Dodanie</u>	$O(1)$	$O(1)$
<u>Usunięcie</u>	$O(1)$	$O(1)$
<u>Wyszukiwanie</u>	$O(n)$	$O(n)$

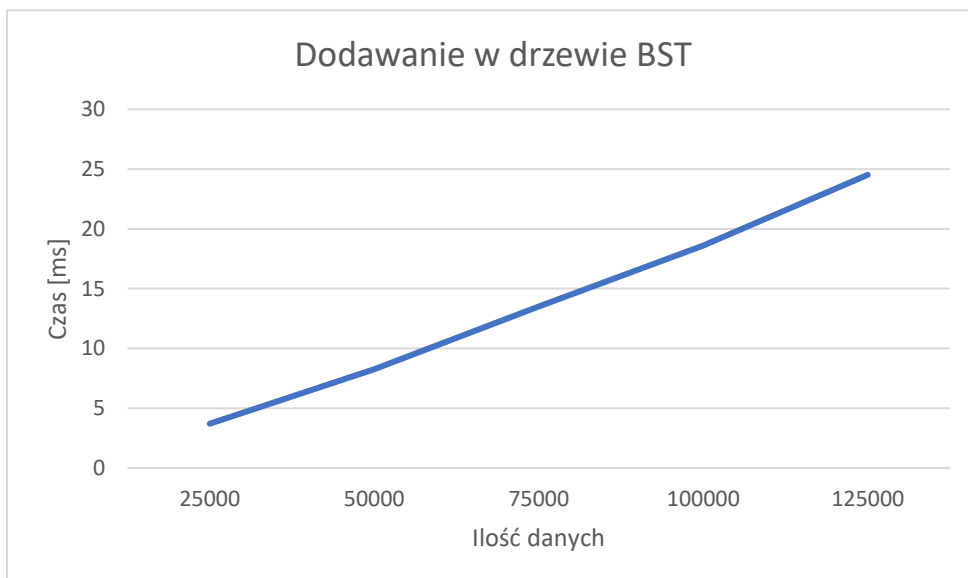
Ilość danych	25000	50000	75000	100000	125000
Dodawanie [ms]	0,56	1,02	1,57	2,04	2,74
Wyszukiwanie [ms]	228,61	862,415	1804,45	3036,95	4464,34
Usuwanie [ms]	0,17	0,37	0,39	0,69	0,86

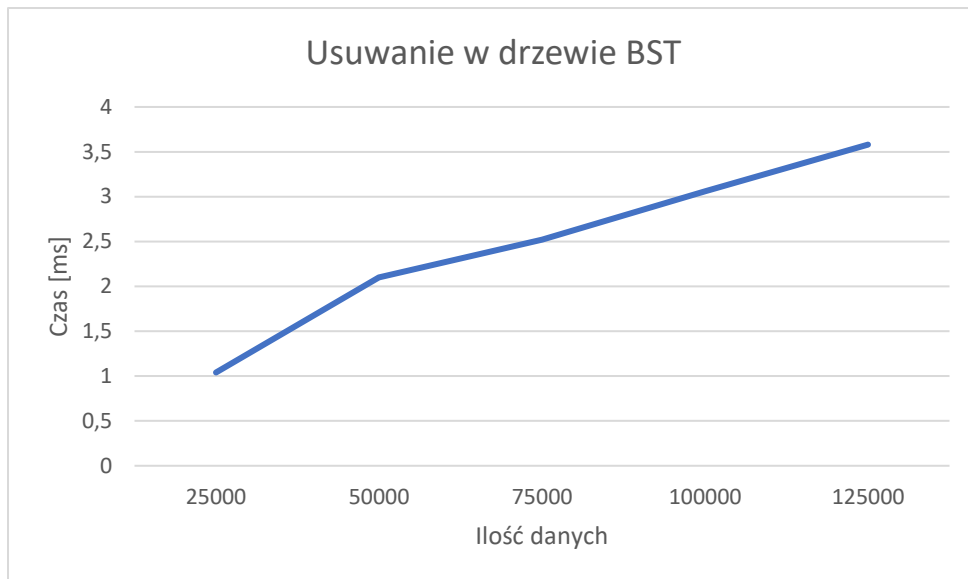


Złożoność drzewa BST:

Funkcja	Średnia	Pesymistyczna
Dodanie	$O(\log(n))$	$O(n)$
Usunięcie	$O(\log(n))$	$O(n)$
Wyszukanie	$O(\log(n))$	$O(n)$

Ilość danych	25000	50000	75000	100000	125000
Dodawanie [ms]	3,71	8,28	13,52	18,6	24,52
Wyszukiwanie [ms]	2,32	5,52	8,84	12,18	15,34
Usuwanie [ms]	1,04	2,1	2,52	3,06	3,58





3. Wnioski

Analizując wyniki można dojść do wniosku, że teoretyczna złożoność obliczeniowa struktur pokrywa się z wynikami. Najmniej efektywną strukturą pod względem złożoności okazała się tablica dynamiczna, z otrzymanych danych, wynika, że złożoność usuwania środkowych wartości w tablicy jest bardziej efektywne niż jakiegokolwiek inne podejście, a dodawanie na koniec tablicy jest znacząco szybsze niż zastosowanie np. dodawania na początek – może to wynikać, ze złej implementacji bądź testowania struktur.

Lista dwukierunkowa osiąga podobne wyniki jak kopiec, lecz jest odrobinę wolniejszy – niezależnie czy usuwamy początek czy koniec listy, kopiec wciąż jest szybszy.

Przeprowadzone zdanie wykazało, że spośród badanych struktur, najszybszą, najefektywniejszą strukturą jest drzewo BST, jest szybsze od każdej wcześniejszej struktury, szczególnie podczas wyszukiwania.

4. Literatura

Thomas Cormen, *Wprowadzenie Do Algorytmów*