
Zadanie projektowe nr 1

Badanie efektywności operacji dodawania, usuwania oraz wyszukiwania elementów
w różnych strukturach danych

Autor sprawozdania: Michał Dziedziak, 263901

Nazwisko i Imię prowadzącego kurs: Mgr. inż. Antoni Sterna

Dzień i godzina zajęć: Wtorek, 15:15 - 16:55 TN

Spis treści

1	Wstęp	3
1.1	Zarys projektu	3
1.2	Struktury danych	3
1.2.1	Zaimplementowane struktury i operacje	3
1.2.2	Teoretyczne złożoności obliczeniowe	3
1.2.3	Złożoność obliczeniowa operacji w tablicy dynamicznej	4
1.2.4	Złożoność obliczeniowa operacji w liście dwukierunkowej	4
1.2.5	Złożoność obliczeniowa operacji w kopcu binarnym	4
1.2.6	Złożoność obliczeniowa operacji w drzewie czerwono-czarnym	4
2	Plan eksperymentu	4
2.1	Założenia	4
2.2	Pomiar czasu	5
2.3	Losowanie populacji	5
3	Zestawienie wyników	5
3.1	Wyniki testów dla tablicy dynamicznej	5
3.2	Wyniki testów dla listy dwukierunkowej	8
3.3	Wyniki testów dla kopca binarnego	11
3.4	Wyniki testów dla drzewa czerwono-czarnego	13
4	Wnioski	14
4.1	Przykładowe wyliczenia	14
4.2	Komentarz	17

Spis tabel

1	Pomiary tablicy dynamicznej 1/2	5
2	Pomiary tablicy dynamicznej 2/2	5
3	Pomiary listy dwukierunkowej 1/2	8
4	Pomiary listy dwukierunkowej 2/2	8

5	Pomiary kopca binarnego	11
6	Pomiary drzewa czerwono-czarnego	13
7	Porównanie otrzymanych czasów dla tablicy dynamicznej z teorią	15
8	Porównanie otrzymanych czasów dla listy dwukierunkowej z teorią	16
9	Porównanie otrzymanych czasów dla kopca binarnego z teorią	17
10	Porównanie otrzymanych czasów dla drzewo czerwono-czarnego z teorią	17

Spis rysunków

1	Wykres czasu od ilości elementów dla Tablicy - dodaj na początek	6
2	Wykres czasu od ilości elementów dla Tablicy - dodaj na koniec	6
3	Wykres czasu od ilości elementów dla Tablicy - dodaj w wybrane miejsce	6
4	Wykres czasu od ilości elementów dla Tablicy - usuń z początku	7
5	Wykres czasu od ilości elementów dla Tablicy - usuń z końca	7
6	Wykres czasu od ilości elementów dla Tablicy - usuń z wybranego miejsca	7
7	Wykres czasu od ilości elementów dla Tablicy - znajdź element	8
8	Wykres czasu od ilości elementów dla Listy - dodaj na początek	9
9	Wykres czasu od ilości elementów dla Listy - dodaj na koniec	9
10	Wykres czasu od ilości elementów dla Listy - dodaj w wybrane miejsce	9
11	Wykres czasu od ilości elementów dla Listy - usuń z początku	10
12	Wykres czasu od ilości elementów dla Listy - usuń z końca	10
13	Wykres czasu od ilości elementów dla Listy - usuń z wybranego miejsca	10
14	Wykres czasu od ilości elementów dla Listy - znajdź element	11
15	Wykres czasu od ilości elementów dla Kopca - dodaj element (push)	11
16	Wykres czasu od ilości elementów dla Kopca - usuń z głowy (pop)	12
17	Wykres czasu od ilości elementów dla Kopca - znajdź element	12
18	Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - dodaj element	13
19	Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - usuń element	13
20	Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - znajdź element	14

1 Wstęp

1.1 Zarys projektu

W ramach projektu w języku C++ zaimplementowane zostały wybrane struktury danych i operacje z nimi związane. Poniższe sprawozdanie bada zależność czasu od ilości elementów w danych strukturach oraz porównuje to za teorią.

1.2 Struktury danych

1.2.1 Zaimplementowane struktury i operacje

- Tablica dynamiczna
 - dodawanie na początek
 - dodawanie na koniec
 - dodawanie w wybrane miejsce
 - usuwanie z początku
 - usuwanie z końca
 - usuwanie z wybranego miejsca
 - szukanie elementu
- Lista dwukierunkowa
 - dodawanie na początek
 - dodawanie na koniec
 - dodawanie w wybrane miejsce
 - usuwanie z początku
 - usuwanie z końca
 - usuwanie z wybranego miejsca
 - szukanie elementu
- Kopiec binarny
 - dodawanie elementu
 - usuwanie głowy
 - szukanie elementu
- Drzewo czerwono-czarne
 - dodawanie elementu
 - usuwanie elementu
 - szukanie elementu

1.2.2 Teoretyczne złożoności obliczeniowe

W tej sekcji przedstawię teoretyczne złożoności obliczeniowe operacji na strukturach, celem późniejszego porównania ich z wynikami testów.

1.2.3 Złożoność obliczeniowa operacji w tablicy dynamicznej

Operacja	Średnia złożoność obliczeniowa
Dodanie elementu na początek	$O(n)$
Dodanie elementu na koniec	$O(n)$
Dodanie elementu w wybrane miejsce	$O(n)$
Usuwanie elementu z początku	$O(n)$
Usuwanie elementu z końca	$O(n)$
Usuwanie elementu z wybranego miejsca	$O(n)$
Szukanie elementu	$O(n)$

1.2.4 Złożoność obliczeniowa operacji w liście dwukierunkowej

Operacja	Średnia złożoność obliczeniowa
Dodanie elementu na początek	$O(1)$
Dodanie elementu na koniec	$O(1)$
Dodanie elementu w wybrane miejsce	$O(n)$
Usuwanie elementu z początku	$O(1)$
Usuwanie elementu z końca	$O(1)$
Usuwanie elementu z wybranego miejsca	$O(n)$
Szukanie elementu	$O(n)$

1.2.5 Złożoność obliczeniowa operacji w kopcu binarnym

Operacja	Średnia złożoność obliczeniowa
Dodanie elementu	$O(\log n)$
Usunięcie głowy	$O(\log n)$
Szukanie elementu	$O(n)$

1.2.6 Złożoność obliczeniowa operacji w drzewie czerwono-czarnym

Operacja	Średnia złożoność obliczeniowa
Dodanie elementu	$O(\log n)$
Usuwanie elementu	$O(\log n)$
Szukanie elementu	$O(\log n)$

2 Plan eksperymentu

2.1 Założenia

- Elementem każdej struktury jest 4 bajtowa liczba całkowita ze znakiem.
- Zakres (wielość) liczby będącej elementem zależy od ilości elementów w strukturze i jest równe przedziałowi od $-\langle \text{liczba elementów} \rangle$ do $+\langle \text{liczba elementów} \rangle$
- Każda struktura jest alokowana dynamicznie, zajmując jak najmniej pamięci (miejsce w strukturach nie jest rezerwowane z zapasem).
- Przeprowadzone testy badają zależność czasu od ilości elementów w strukturze.
- Dla każdej operacji test powtarzany jest sto razy. Podczas każdej próby losowana jest nowa populacja, a ostateczny czas jest średnią czasów z prób.

- Testy rozpoczynają się dla 100000 elementów w strukturze i są zwiększane liniowo o 100000, aż do osiągnięcia 1000000 elementów.
- Dla każdego testu najpierw dodawana jest odpowiednia ilość elementów do struktury, a następnie mierzony jest czas wykonywania danej operacji.

2.2 Pomiar czasu

Do pomiaru czasu została napisana osobna klasa "Timer". Korzysta ona z funkcji QueryPerformanceCounter i QueryPerformanceFrequency umożliwiających pomiar czasu z dokładnością do mikro sekund. Klasa ta określa upływ czasu, bazując na dokładnym liczniku.

2.3 Losowanie populacji

Do losowania populacji również została napisana osobna klasa RandomGenerator. Umożliwia ona losowanie pojedynczych liczb lub całych zbiorów. Korzysta z mt19937 - generatora liczb pseudolosowych.

3 Zestawienie wyników

3.1 Wyniki testów dla tablicy dynamicznej

Ilość elementów	Czas potrzebny na wykonanie operacji [ms]		
	Dodaj na początek	Dodaj na koniec	Dodaj w wybrane miejsce
100000	0.31	0.27	0.28
200000	0.69	0.60	0.69
300000	1.18	1.30	1.17
400000	1.46	1.63	1.46
500000	1.79	2.20	1.99
600000	2.14	2.19	2.27
700000	2.33	2.30	2.33
800000	2.83	2.98	3.05
900000	3.01	2.94	3.41
1000000	3.70	3.34	3.66

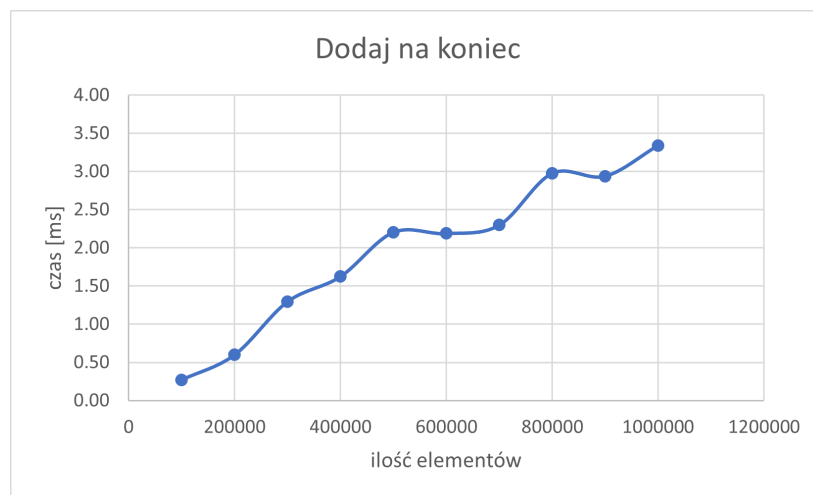
Tabela 1: Pomiary tablicy dynamicznej 1/2

Ilość elementów	Czas potrzebny na wykonanie operacji [ms]			
	Usuń z początku	Usuń z końca	Usuń z wybranego miejsca	Znajdź element
100000	0.25	0.26	0.25	0.20
200000	0.66	0.62	0.62	0.30
300000	1.02	1.18	1.13	0.51
400000	1.57	1.46	1.30	0.67
500000	1.92	1.84	1.63	0.83
600000	2.15	2.05	1.95	1.01
700000	2.63	2.41	2.52	1.37
800000	2.56	2.66	2.86	1.53
900000	3.17	3.04	3.23	1.47
1000000	3.71	3.49	3.54	1.82

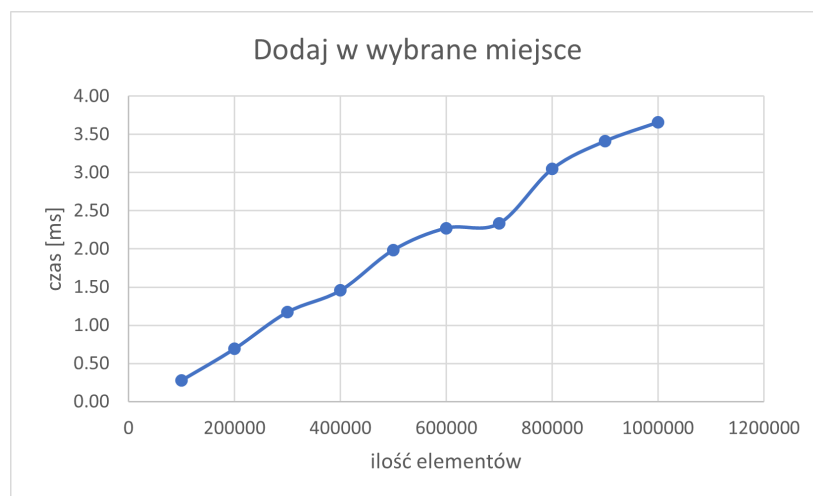
Tabela 2: Pomiary tablicy dynamicznej 2/2



Rysunek 1: Wykres czasu od ilości elementów dla Tablicy - dodaj na początek



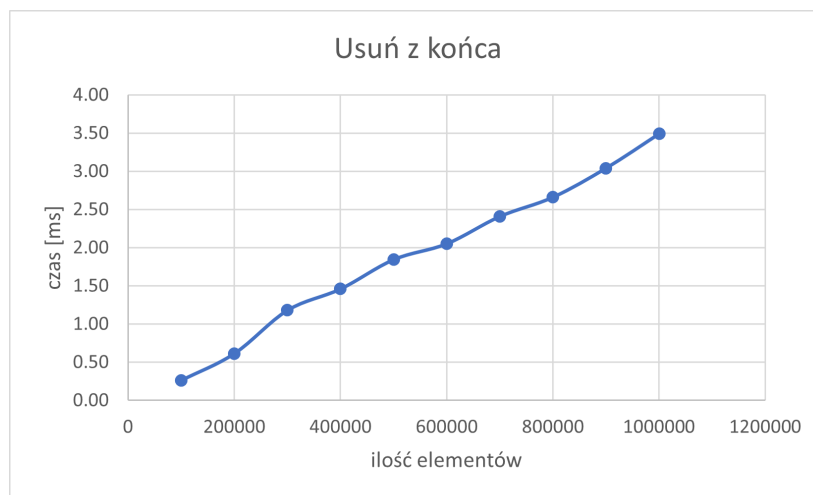
Rysunek 2: Wykres czasu od ilości elementów dla Tablicy - dodaj na koniec



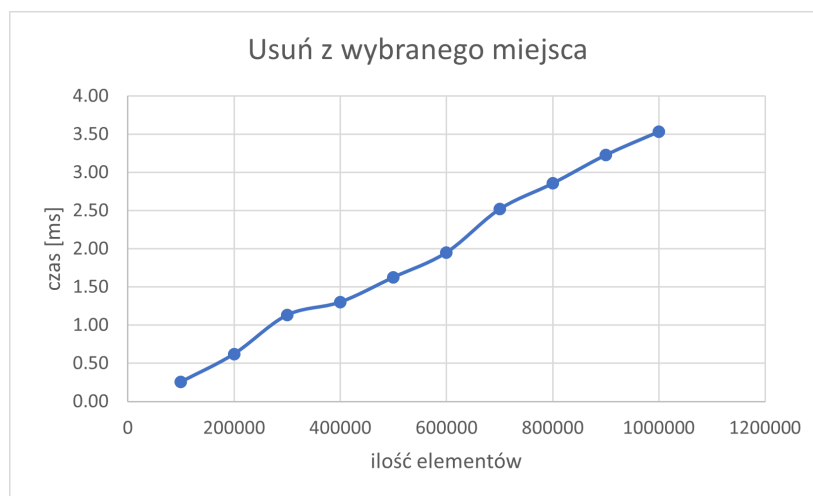
Rysunek 3: Wykres czasu od ilości elementów dla Tablicy - dodaj w wybrane miejsce



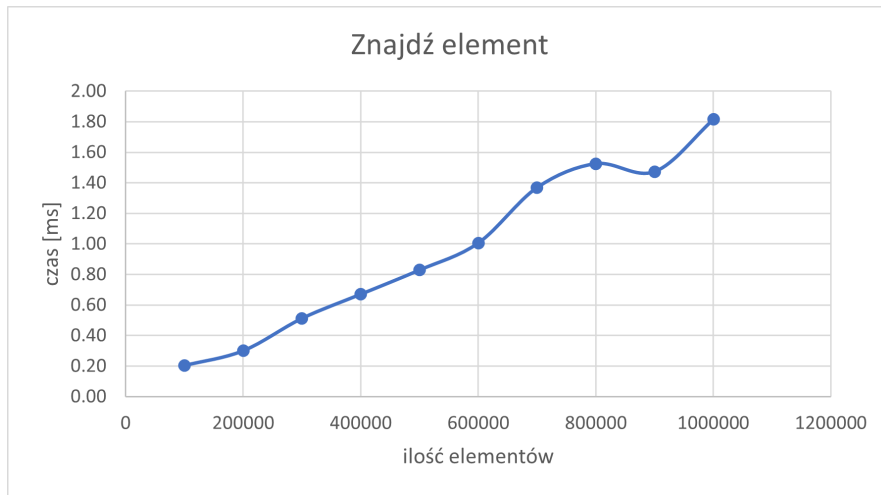
Rysunek 4: Wykres czasu od ilości elementów dla Tablicy - usuń z początku



Rysunek 5: Wykres czasu od ilości elementów dla Tablicy - usuń z końca



Rysunek 6: Wykres czasu od ilości elementów dla Tablicy - usuń z wybranego miejsca



Rysunek 7: Wykres czasu od ilości elementów dla Tablicy - znajdź element

3.2 Wyniki testów dla listy dwukierunkowej

Ilość elementów	Czas potrzebny na wykonanie operacji [us]			
	Dodaj na początek	Dodaj na koniec	Usuń z początku	Usuń z końca
100000	0.11	0.08	0.72	0.52
200000	0.14	0.10	0.74	0.56
300000	0.13	0.10	0.81	0.50
400000	0.12	0.13	0.89	0.50
500000	0.14	0.11	0.74	0.47
600000	0.11	0.10	0.73	0.46
700000	0.12	0.09	0.63	0.49
800000	0.12	0.09	0.66	0.48
900000	0.12	0.09	0.65	0.49
1000000	0.12	0.09	0.66	0.46

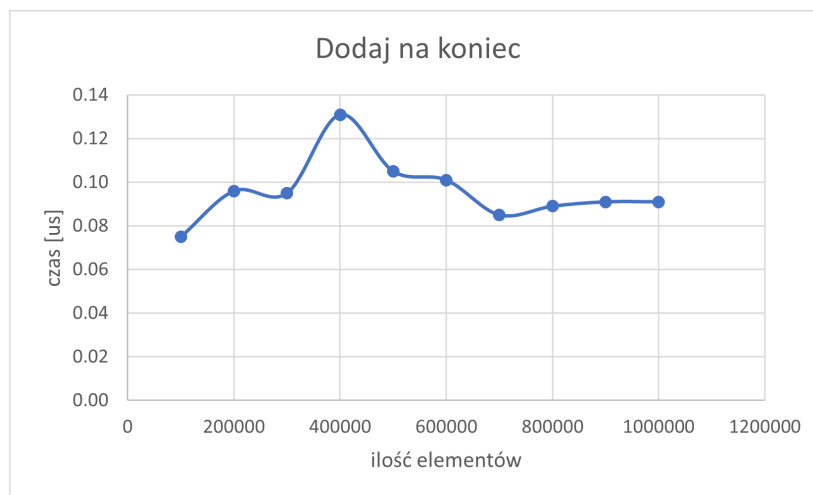
Tabela 3: Pomiary listy dwukierunkowej 1/2

Ilość elementów	Czas potrzebny na wykonanie operacji [us]		
	Dodaj w wybrane miejsce	Usuń z wybranego miejsca	Znajdź element
100000	0.23	0.26	0.72
200000	0.49	0.50	1.40
300000	0.69	0.77	2.07
400000	0.97	1.16	2.44
500000	1.33	1.21	3.47
600000	1.70	1.61	4.62
700000	1.76	1.85	4.75
800000	2.20	2.11	5.70
900000	2.15	2.22	6.19
1000000	2.72	2.47	6.94

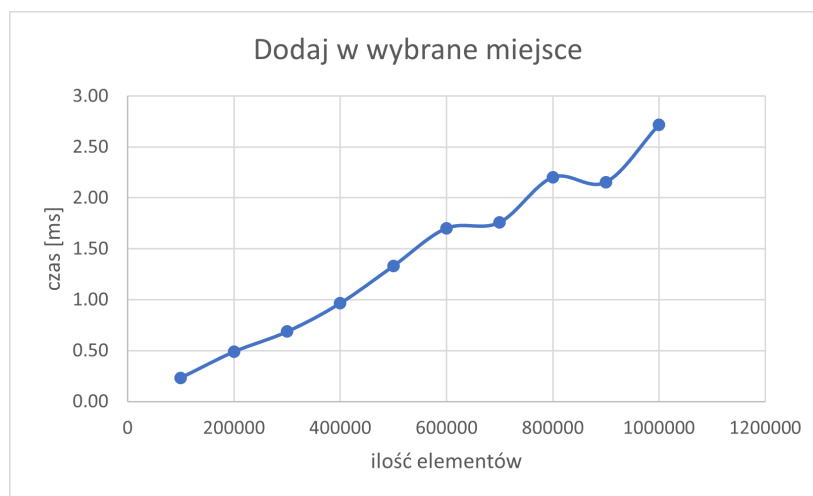
Tabela 4: Pomiary listy dwukierunkowej 2/2



Rysunek 8: Wykres czasu od ilości elementów dla Listy - dodaj na początek



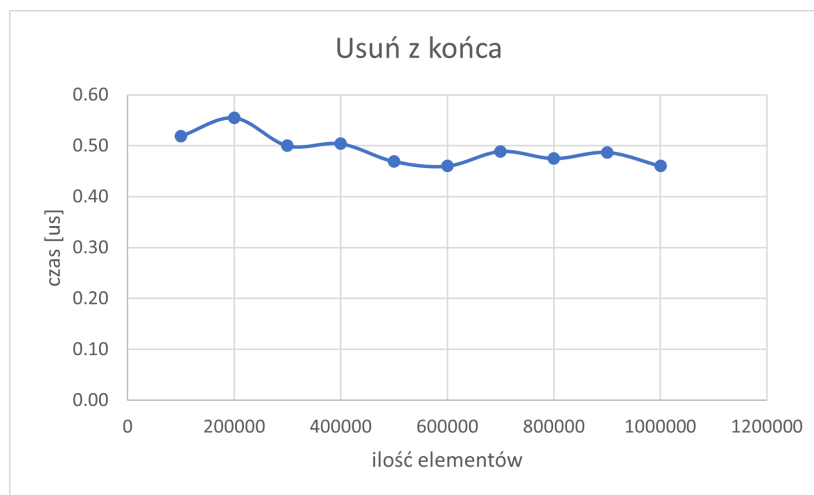
Rysunek 9: Wykres czasu od ilości elementów dla Listy - dodaj na koniec



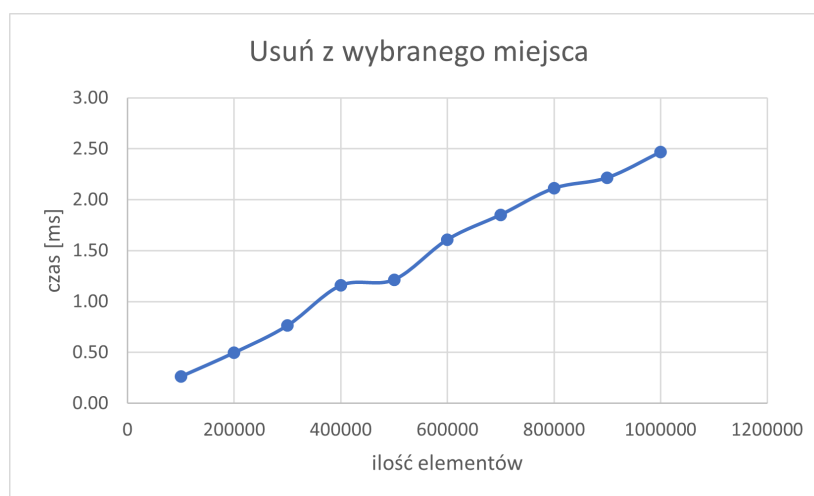
Rysunek 10: Wykres czasu od ilości elementów dla Listy - dodaj w wybrane miejsce



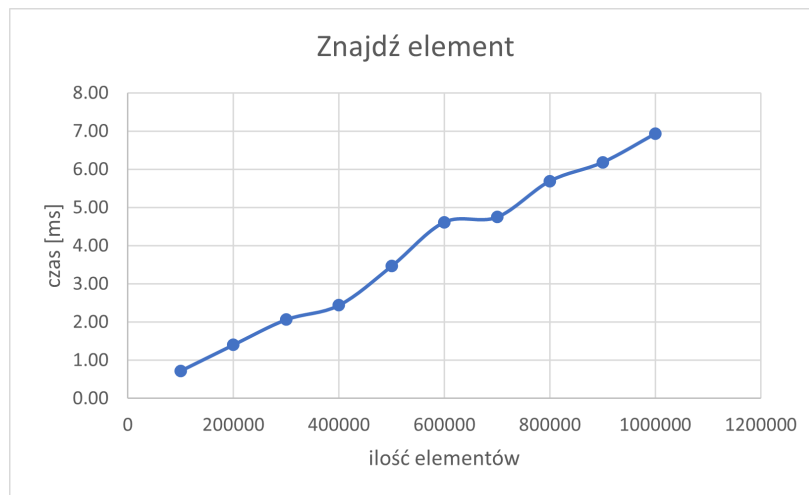
Rysunek 11: Wykres czasu od ilości elementów dla Listy - usuń z początku



Rysunek 12: Wykres czasu od ilości elementów dla Listy - usuń z końca



Rysunek 13: Wykres czasu od ilości elementów dla Listy - usuń z wybranego miejsca

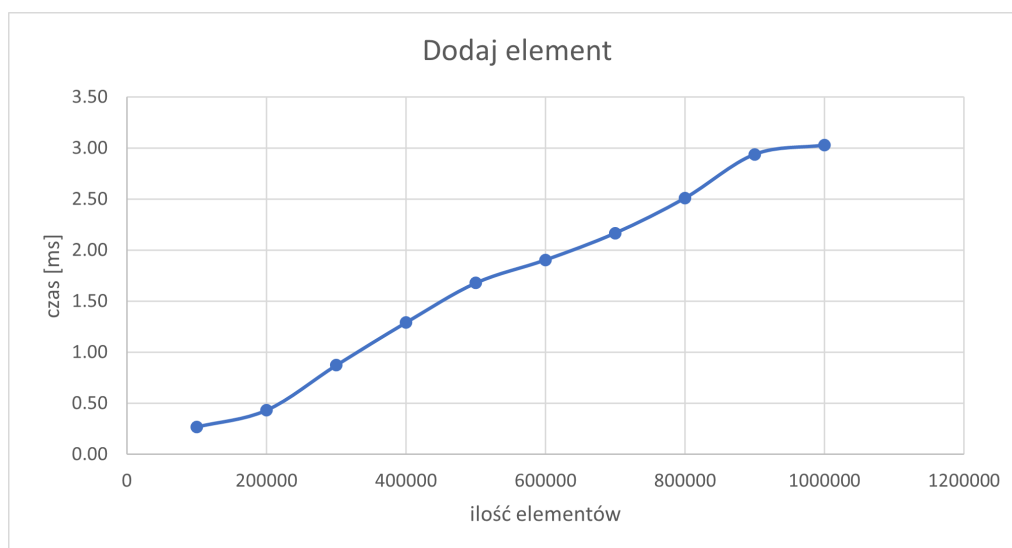


Rysunek 14: Wykres czasu od ilości elementów dla Listy - znajdź element

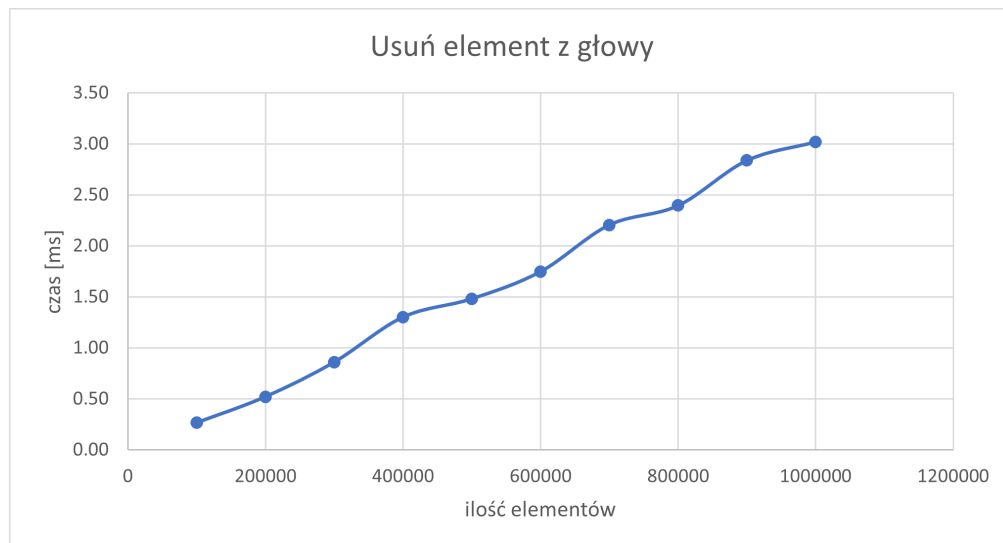
3.3 Wyniki testów dla kopca binarnego

Ilość elementów	Czas potrzebny na wykonanie operacji [ms]		
	Dodaj element	Usuń element z głowy	Znajdź element
100000	0.27	0.26	0.15
200000	0.43	0.52	0.35
300000	0.87	0.86	0.45
400000	1.29	1.30	0.63
500000	1.68	1.48	0.88
600000	1.90	1.75	1.05
700000	2.17	2.20	1.28
800000	2.51	2.40	1.55
900000	2.94	2.84	1.48
1000000	3.03	3.02	1.82

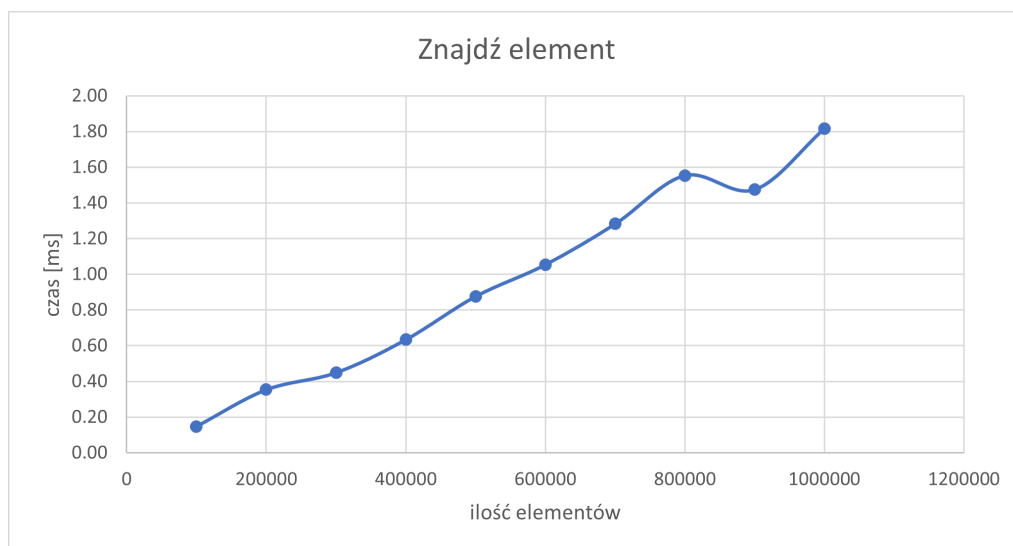
Tabela 5: Pomiary kopca binarnego



Rysunek 15: Wykres czasu od ilości elementów dla Kopca - dodaj element (push)



Rysunek 16: Wykres czasu od ilości elementów dla Kopca - usuń z głowy (pop)

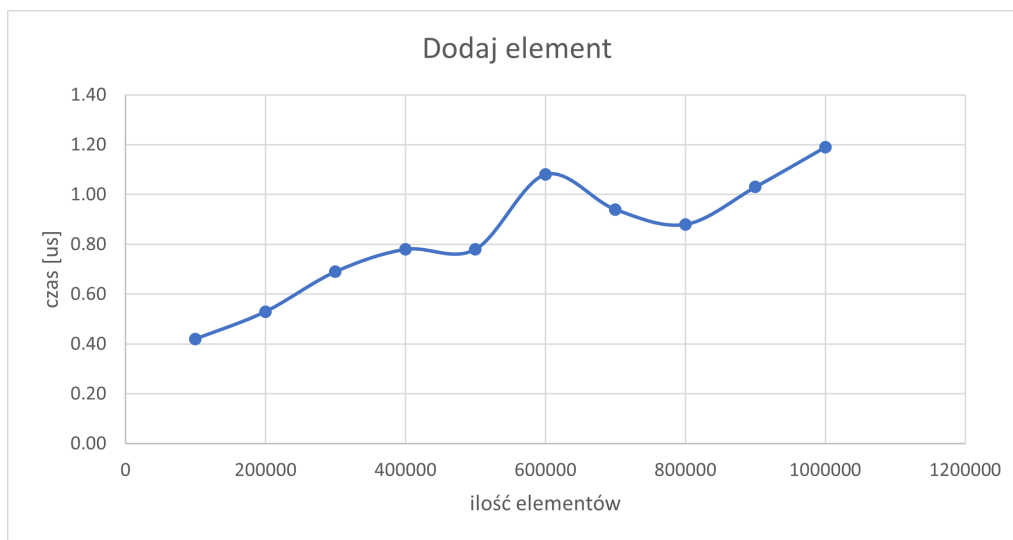


Rysunek 17: Wykres czasu od ilości elementów dla Kopca - znajdź element

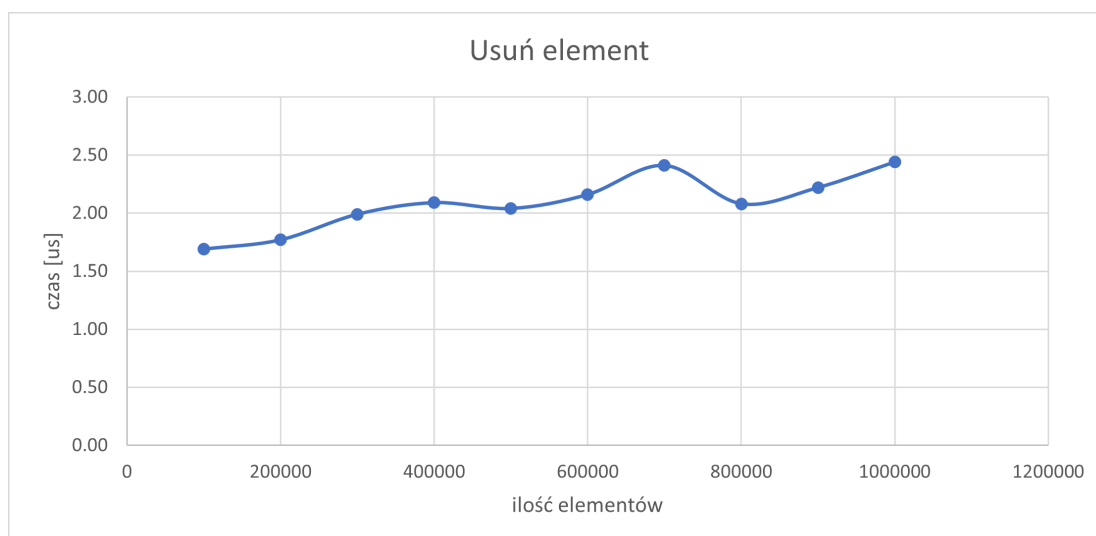
3.4 Wyniki testów dla drzewa czerwono-czarnego

Ilość elementów	Czas potrzebny na wykonanie operacji [us]		
	Dodaj element	Usuń element	Znajdź element
100000	0.42	1.69	0.27
200000	0.53	1.77	0.47
300000	0.69	1.99	0.60
400000	0.78	2.09	0.54
500000	0.78	2.04	0.64
600000	1.08	2.16	0.84
700000	0.94	2.41	0.67
800000	0.88	2.08	0.80
900000	1.03	2.22	0.76
1000000	1.19	2.44	0.81

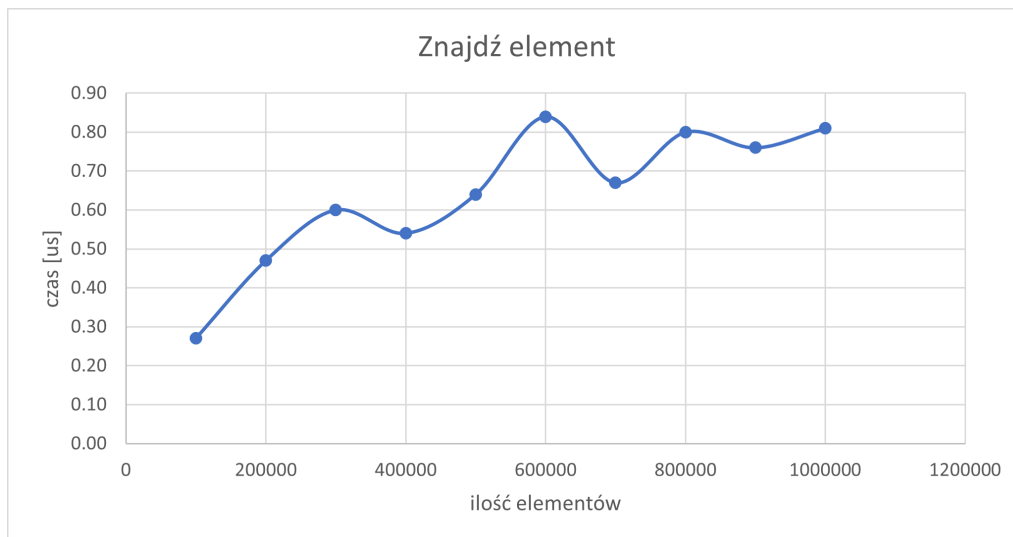
Tabela 6: Pomiary drzewa czerwono-czarnego



Rysunek 18: Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - dodaj element



Rysunek 19: Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - usuń element



Rysunek 20: Wykres czasu od ilości elementów dla Drzewa czerwono-czarnego - znajdź element

4 Wnioski

4.1 Przykładowe wyliczenia

W tej sekcji porównane zostaną czasy otrzymane eksperymentalnie z tymi wynikającymi z teorii. Porównywane będą zależności czasu wykonania danych operacji dla rozmiarów:

- 400 tysięcy i 100 tysięcy (proporcja 4:1)
- 800 tysięcy i 100 tysięcy (proporcja 8:1)
- 800 tysięcy i 200 tysięcy (proporcja 4:1)

Tablica dynamiczna

Wybrane wyliczenia porównujące czasy operacji tablicy dynamicznej z teorią						
Rodzaj operacji	Teoretyczna złożoność	Ilość elementów	Czas otrzymany eksperymentalnie	Stosunek ilości elementów	Teoretyczny czas wykonania operacji	Różnica
Dodaj na początek	O(n)	100000	0.31	4	X	X
		400000	1.46		1.25	15%
		100000	0.31		X	X
		800000	2.83	8	2.50	12%
		200000	0.69		X	X
		800000	2.83		2.75	3%
Dodaj na koniec	O(n)	100000	0.27	4	X	X
		400000	1.63		1.09	33%
		100000	0.27		X	X
		800000	2.98	8	2.19	26%
		200000	0.60		X	X
		800000	2.98		2.40	19%
Dodaj w wybrane miejsce	O(n)	100000	0.28	4	X	X
		400000	1.46		1.12	23%
		100000	0.28		X	X
		800000	3.05	8	2.24	26%
		200000	0.69		X	X
		800000	3.05		2.77	9%
Usuń z początku	O(n)	100000	0.25	4	X	X
		400000	1.57		0.99	37%
		100000	0.25		X	X
		800000	2.56	8	1.98	23%
		200000	0.66		X	X
		800000	2.56		2.64	3%
Usuń z końca	O(n)	100000	0.26	4	X	X
		400000	1.46		1.06	27%
		100000	0.26		X	X
		800000	2.66	8	2.12	20%
		200000	0.62		X	X
		800000	2.66		2.46	7%
Usuń z wybranego miejsca	O(n)	100000	0.25	4	X	X
		400000	1.30		1.01	22%
		100000	0.25		X	X
		800000	2.86	8	2.03	29%
		200000	0.62		X	X
		800000	2.86		2.48	13%
Znajdź element	O(n)	100000	0.20	4	X	X
		400000	0.67		0.82	18%
		100000	0.20		X	X
		800000	1.53	8	1.64	7%
		200000	0.30		X	X
		800000	1.53		1.21	21%

Tabela 7: Porównanie otrzymanych czasów dla tablicy dynamicznej z teorią

Lista dwukierunkowa

Wybrane wyliczenia porównujące czasy operacji listy dwukierunkowej z teorią						
Rodzaj operacji	Teoretyczna złożoność	Ilość elementów	Czas otrzymany eksperymentalnie	Stosunek ilości elementów	Teoretyczny czas wykonania operacji	Różnica
Dodaj na początek	O(1)	100000	0.11	4	X	X
		400000	0.12		0.11	7%
		100000	0.11	8	X	X
		800000	0.12		0.11	7%
		200000	0.14	4	X	X
		800000	0.12		0.14	19%
Dodaj na koniec	O(1)	100000	0.08	4	X	X
		400000	0.13		0.08	43%
		100000	0.08	8	X	X
		800000	0.09		0.08	16%
		200000	0.10	4	X	X
		800000	0.09		0.10	7%
Usuń z początku	O(1)	100000	0.72	4	X	X
		400000	0.89		0.72	19%
		100000	0.72	8	X	X
		800000	0.66		0.72	8%
		200000	0.74	4	X	X
		800000	0.66		0.74	11%
Usuń z końca	O(1)	100000	0.52	4	X	X
		400000	0.50		0.52	3%
		100000	0.52	8	X	X
		800000	0.48		0.52	8%
		200000	0.56	4	X	X
		800000	0.48		0.56	14%
Dodaj w wybrane miejsce	O(n)	100000	0.23	4	X	X
		400000	0.97		0.93	3%
		100000	0.23	8	X	X
		800000	2.20		1.87	15%
		200000	0.49	4	X	X
		800000	2.20		1.96	11%
Usuń z wybranego miejsca	O(n)	100000	0.26	4	X	X
		400000	1.16		1.06	9%
		100000	0.26	8	X	X
		800000	2.11		2.11	0%
		200000	0.50	4	X	X
		800000	2.11		1.99	6%
Znajdź element	O(n)	100000	0.72	4	X	X
		400000	2.44		2.89	16%
		100000	0.72	8	X	X
		800000	5.70		5.78	2%
		200000	1.40	4	X	X
		800000	5.70		5.62	1%

Tabela 8: Porównanie otrzymanych czasów dla listy dwukierunkowej z teorią

Kopiec binarny

Wybrane wyliczenia porównujące czasy operacji kopca binarnego z teorią						
Rodzaj operacji	Teoretyczna złożoność	Ilość elementów	Czas otrzymany eksperymentalnie	Stosunek ilości elementów	Teoretyczny czas wykonania operacji	Różnica
Dodaj element	$O(\log n)$	100000	0.27	4	X	X
		400000	1.29		0.54	58%
		100000	0.27		X	X
		800000	2.51	8	0.81	68%
		200000	0.43	4	X	X
		800000	2.51		0.87	66%
Usuń element z głowy	$O(\log n)$	100000	0.26	4	X	X
		400000	1.30		0.53	59%
		100000	0.26	8	X	X
		800000	2.40		0.79	67%
		200000	0.52	4	X	X
		800000	2.40		1.04	57%
Znajdź element	$O(n)$	100000	0.15	4	X	X
		400000	0.63		0.59	7%
		100000	0.15	8	X	X
		800000	1.55		1.18	24%
		200000	0.35	4	X	X
		800000	1.55		1.42	9%

Tabela 9: Porównanie otrzymanych czasów dla kopca binarnego z teorią

Drzewo czerwono-czarne

Wybrane wyliczenia porównujące czasy operacji drzewa czerwono-czarnego z teorią						
Rodzaj operacji	Teoretyczna złożoność	Ilość elementów	Czas otrzymany eksperymentalnie	Stosunek ilości elementów	Teoretyczny czas wykonania operacji	Różnica
Dodaj element	$O(\log n)$	100000	0.42	4	X	X
		400000	0.78		0.84	7%
		100000	0.42		X	X
		800000	0.88	8	1.26	30%
		200000	0.53	4	X	X
		800000	0.88		1.06	17%
Usuń element	$O(\log n)$	100000	1.69	4	X	X
		400000	2.09		3.38	38%
		100000	1.69	8	X	X
		800000	2.08		5.07	59%
		200000	1.77	4	X	X
		800000	2.08		3.54	41%
Znajdź element	$O(\log n)$	100000	0.27	4	X	X
		400000	0.54		0.54	0%
		100000	0.27	8	X	X
		800000	0.80		0.81	1%
		200000	0.47	4	X	X
		800000	0.80		0.94	15%

Tabela 10: Porównanie otrzymanych czasów dla drzewa czerwono-czarnego z teorią

4.2 Komentarz

Tablica dynamiczna

Wyniki testów przeprowadzonych na tablicy pokrywają się z oczekiwaniami, tzn. otrzymane złożoności odpowiadają tym teoretycznym. Dla zaimplementowanych operacji na tablicy można to łatwo zweryfikować, ponieważ czas rośnie liniowo wraz ze wzrostem ilości elementów w strukturze.

Dodatkowo warto zaznaczyć dwie rzeczy. Po pierwsze, przeprowadzone operacje nie badały czasu odczytu elementu na danej (znanej) pozycji, a jest to główna zaleta tablic. Po drugie dzisiejsze implementacje tablic dynamicznych

wykorzystują wiele technik, które znacząco przyspieszają operacje na powyższej strukturze (np. rezerwowanie miejsca w pamięci na zapas).

Lista dwukierunkowa

Lista dwukierunkowa charakteryzuje się szybkim i stałym czasowo dostępem do pierwszego i ostatniego elementu struktury. W testach dla usuwania/dodawania na początku/końcu można zauważyć drobne wahania czasu w zależności od wielkości struktury, nie przejawiają one jednak żadnego widocznego trendu i wynikają prawdopodobnie z bardzo krótkich czasów tych operacji. Dodatkowo warto zauważyć, że dodawanie/usuwanie elementu o wybranym indeksie i wyszukiwanie elementu cechuje liniowa złożoność obliczeniowa, co również zostało potwierdzone doświadczalnie dla zaimplementowanej struktury.

Kopiec binarny

Kopiec binarny to struktura, która zapewnia, że pierwszy element (tzw. głowa) jest zawsze największym/najmniejszym elementem struktury. Zaimplementowana została wersja w wariancie maksymalnym.

Teoretyczna złożoność obliczeniowa dla dodania elementu i usunięcia głowy to $O(\log n)$ niestety nie pokrywa się ona z tą wykazaną w testach, której bliżej jest do zależności liniowej. Jest to spowodowane tym, że w zastosowanej implementacji dodanie elementu do kopca jest równoznaczne z potrzebą poszerzenia tablicy, w której przechowywane są elementy. Poszerzenie tablicy wymaga z kolei przeniesienia elementów z poprzedniej tablicy do nowej co ma złożoność $O(n)$. Czas zmierzony podczas testów jest sumą czasów tych dwóch operacji co ostatecznie daje złożoność równą $O(n + \log n) = O(n)$.

Zbadana została również operacja szukania elementu. Co do implementacji to jest ona bliźniacza do wyszukiwania w tablicy dynamicznej z tą różnicą, że na początku sprawdzane jest, czy szukany element nie jest większy od głowy - jeżeli tak, element nie znajduje się w kopcu. Testowane były również wariacje sprawdzające, czy na danym poziomie wszystkie wartości nie są mniejsze od szukanego elementu (co również kończyłoby szukanie), lecz w testach były one średnio dużo wolniejsze. Ostatecznie, pomimo usprawnień wyszukiwanie w kopcu dalej ma złożoność teoretyczną $O(n)$ co pokrywa się z otrzymanymi pomiarami.

Warto jeszcze zwrócić uwagę na to, że otrzymane czasy wyszukiwania w kopcu są podobne do czasów wyszukiwania w tablicy. Teoretycznie czasy wyszukiwania w kopcu powinny być średnio krótsze niż w tablicy. Powodem, przez który nie odzwierciedlają tego moje pomiary, może być za mała rozpiętość wielkości liczb w strukturze.

Drzewo czerwono czarne

Drzewo czerwono czarne to rodzaj samobalansującego się drzewa. Zapewnia ono, że operacja wyszukania zawsze ma złożoność $O(\log n)$ co zostało wykazane doświadczalnie. Operacja dodawania i usuwania elementu również ma teoretyczną złożoność obliczeniową równą $O(\log n)$, jak wykazały testy, zaimplementowana struktura osiąga czasy zgodne z przewidywaniami teoretycznymi.