Struktury Danych i Złożoność Obliczeniowa

Sprawozdanie z projektu

Paul Paczyński

Nr albumu: 254307

28.05.2022

Zadanie projektowe nr 1:

Badanie efektywności operacji na danych w podstawowych strukturach danych

Prowadzący kurs: Dr inż. Dariusz Banasiak

Termin zajęć: Piątek 11:15 TN

Termin oddania zadania: 29.04.2022

Spis treści

1.Wstęp

Zadaniem projektowym było napisanie programu i zmierzenie nim czasu wykonania funkcji dodawania, usuwania i wyszukiwania elementu o podanej wartości dla zadanych struktur danych.

Zadane struktury:

• Tablica dynamiczna

• Lista dwukierunkowa

• Kopiec binarny typu maksimum

• Drzewo czerwono-czarne

Dodatkowe założenia wynikające z wytycznych zadania projektowego:

- Typ zmiennej przetrzymywanej w strukturach to int
- Wszystkie struktury są alokowane dynamicznie i po ich zużyciu zwalniana jest pamięć
- Tablica oraz lista posiadają dodatkowe pomiary dodawania na początek i koniec
- Program został napisany w C++ bez gotowych bibliotek ze strukturami
- Dane których używam w pomiarze losowane są z całego zakresu inta
- Pomiary wykonano dla następujących wielkości struktur: 1000, 2000, 3000, 4000, 5000, 10’000, 15’000, 20’000
- Dla każdej grupy losowano zestaw danych sto razy. Każda operacja na jednym ze stu zestawów dodatkowo była powtarzana 10 razy, aby nieco zmniejszyć błąd wynikający z losowości danych

2. Złożoność obliczeniowa struktur

2.1 Złożoność tablicy dynamicznej

Funkcja	*Średnia*	*Pesymistyczna*
Dodanie elementu	*O(n)*	*O(n)*
Usunięcie elementu	*O(n)*	*O(n)*
Wyszukanie elementu	*O(n)*	*O(n)*
Dostęp do elementu	*O(1)*	*O(1)*

2.2 Złożoność listy dwukierunkowej

Funkcja	*Średnia*	*Pesymistyczna*
Dodanie elementu	*O(-)*	*O(n)*
Usunięcie elementu	*O(-)*	*O(n)*
Wyszukanie elementu	*O(n)*	*O(n)*
Dostęp do elementu	*O(n)*	*O(n)*

2.3 Złożoność kopca binarnego

Funkcja	*Średnia*	*Pesymistyczna*
Dodanie elementu	*O(1)*	*O(1)*
Usunięcie elementu	*O(1)*	*O(1)*
Wyszukanie elementu	*O(n)*	*O(n)*
Dostęp do elementu	*O(n)*	*O(n)*

2.4 Złożoność drzewa czerwono czarnego

Funkcja	*Średnia*	*Pesymistyczna*
Dodanie elementu	*O(log(n))*	*O(log(n))*
Usunięcie elementu	*O(log(n))*	*O(log(n))*
Wyszukanie elementu	*O(log(n))*	*O(log(n))*
Dostęp do elementu	*O(log(n))*	*O(log(n))*

3. Opis projektu

W programie zostały zaimplementowane wszystkie obowiązkowe struktury z zadania projektowego. Program pozwala dla każdej struktury sprawdzić jej działanie manualnie. Operacje wyszukania, dodania, usunięcie elementu są dostępne w menu konsolowym. Dodatkowo program posiada tryb automatyczny, służący do wywołania klasy odpowiedzialnej za pomiar czasowy wszystkich struktur dla podanych wielkości grup.

Wyniki przestawione w tym sprawozdaniu zostały zmierzone dla każdej struktury z następującymi parametrami:

Ilość elementów w każdej strukturze: 1000 , 2000, 3000, 4000, 5000, 10000, 15000, 20000

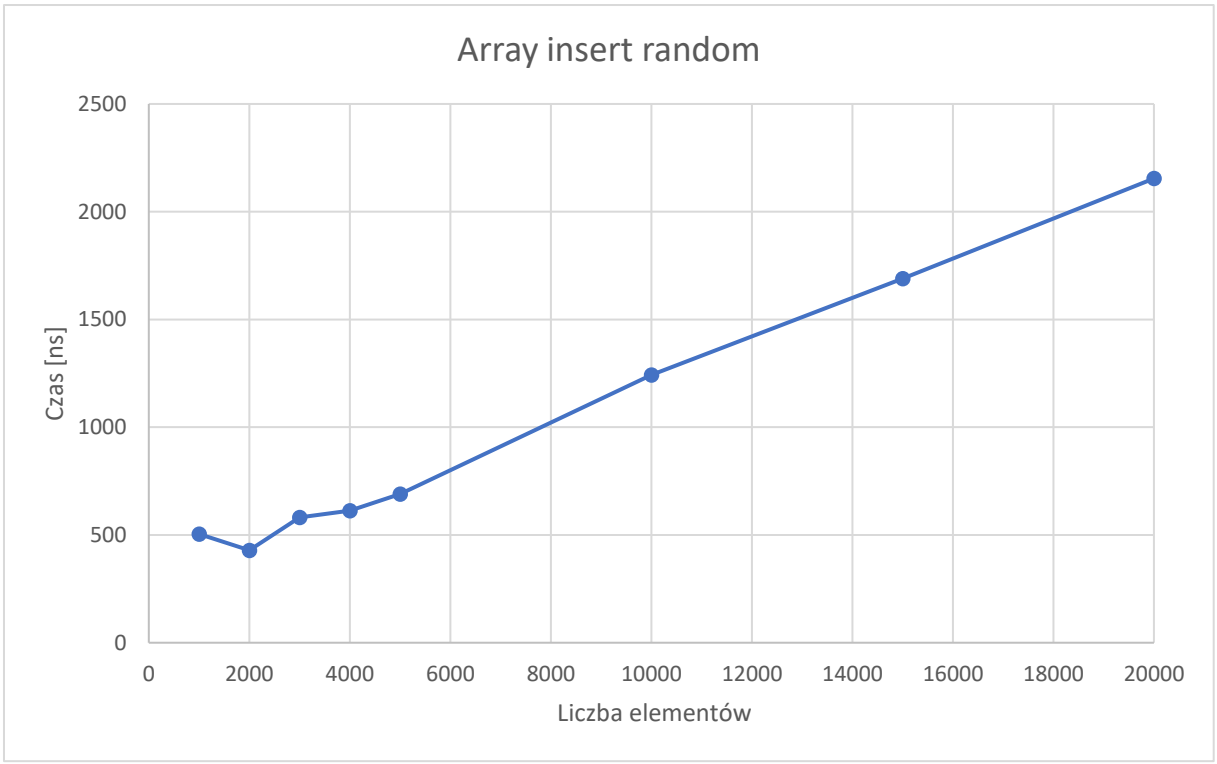
Pomiar został wykonany w następujący sposób:

Program dla każdej struktury tworzy grupę losowych liczb z zakresu całego inta. Powtarza operację losowania dla jednej struktury 100 razy. Dodatkowo dla każdego wylosowanego zbioru powtarza daną operację 10 razy, operujących na tym samym zbiorze. Czas mierzony jest tylko i wyłącznie w momencie wykonania mierzonej operacji. Losowanie liczb nie wpływa na czas wykonania pomiaru. Wyniki zawsze podane są w ns.

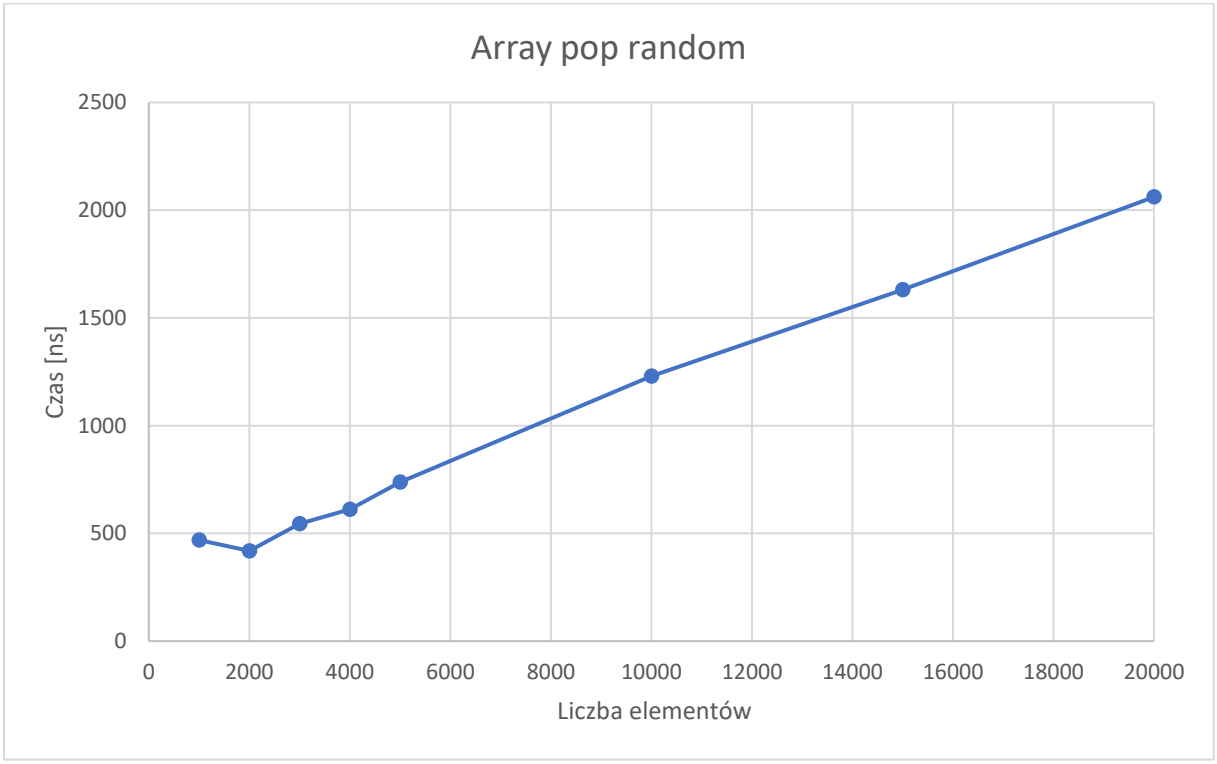
4. Wyniki eksperymentów

4.1 Tablica

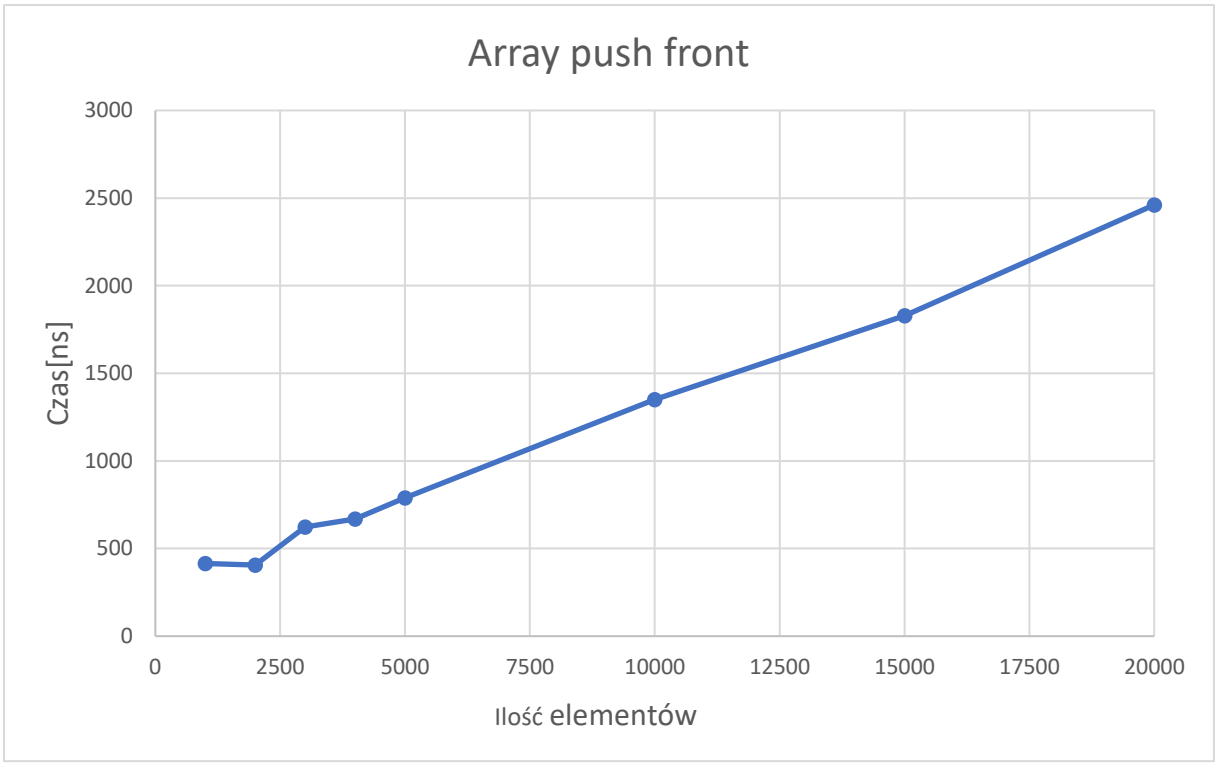
Liczba Elementów	**Push Front**	**Push Back**	**Pop front**	**Pop back**	**Search**	**Push**	**Pop**
	[ns]	***[ns]***	***[ns]***	***[ns]***	***[ns]***	***[ns]***	***[ns]***
1000	415	431	383	356	1188	505	469
2000	406	397	395	413	2148	429	419
3000	623	507	458	458	3558	581	545
4000	668	537	525	528	4182	612	612
5000	789	779	721	708	5245	690	738
10000	1351	1234	1180	1116	10315	1242	1229
15000	1829	1641	1712	1766	15904	1689	1631
20000	2461	2144	2173	2174	22460	2154	2061



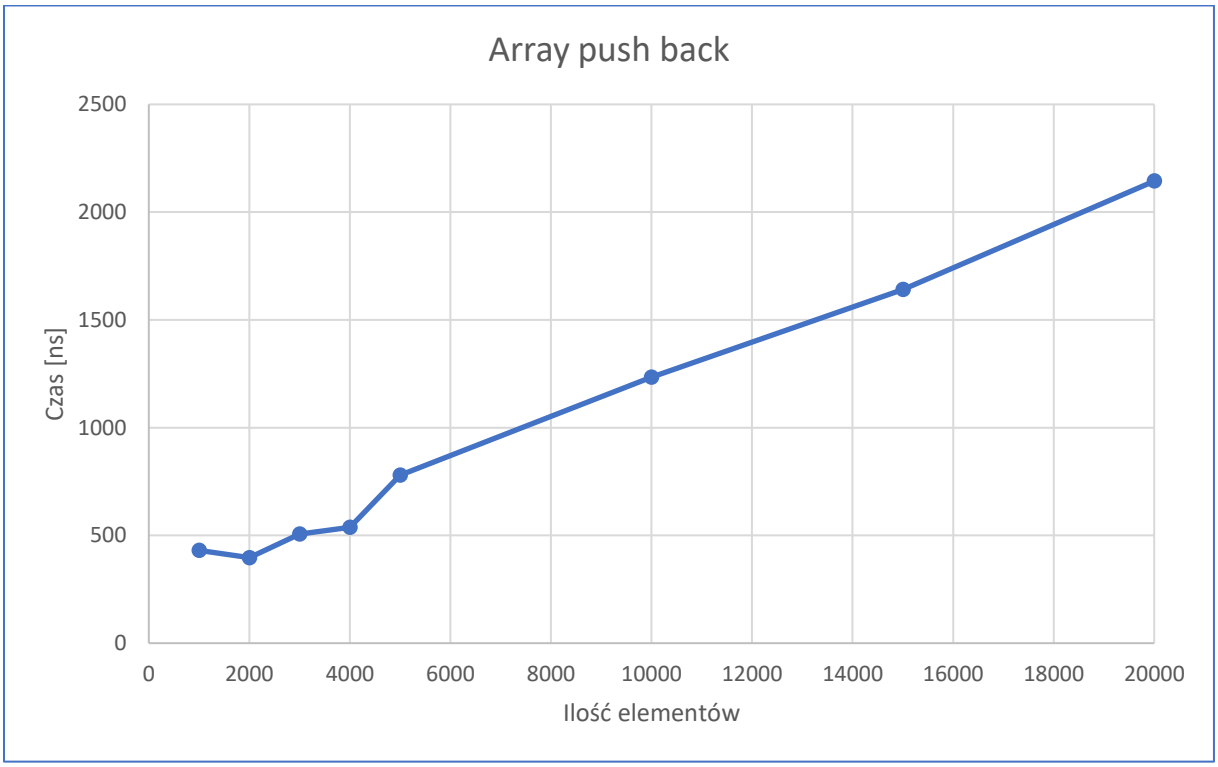
1. Dodanie elementu na losowe miejsce w tablicy



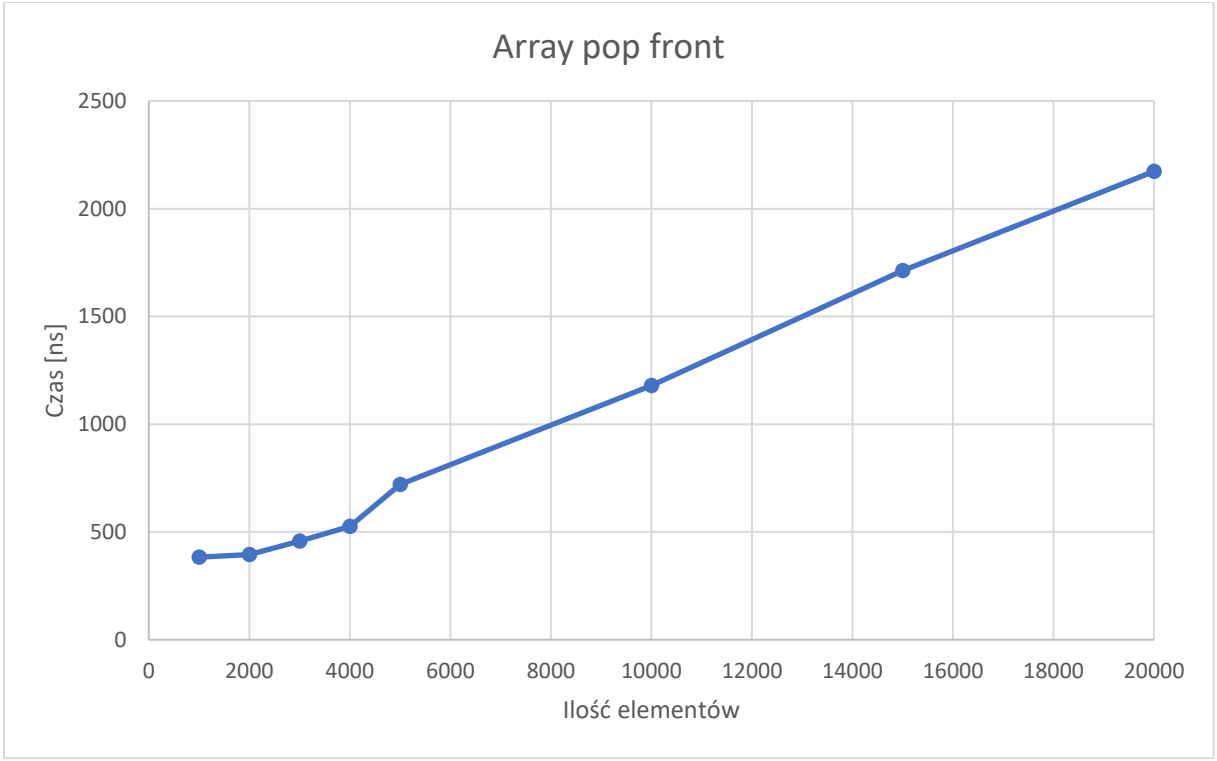
2. Usunięcie elementu z losowego miejsca tablicy



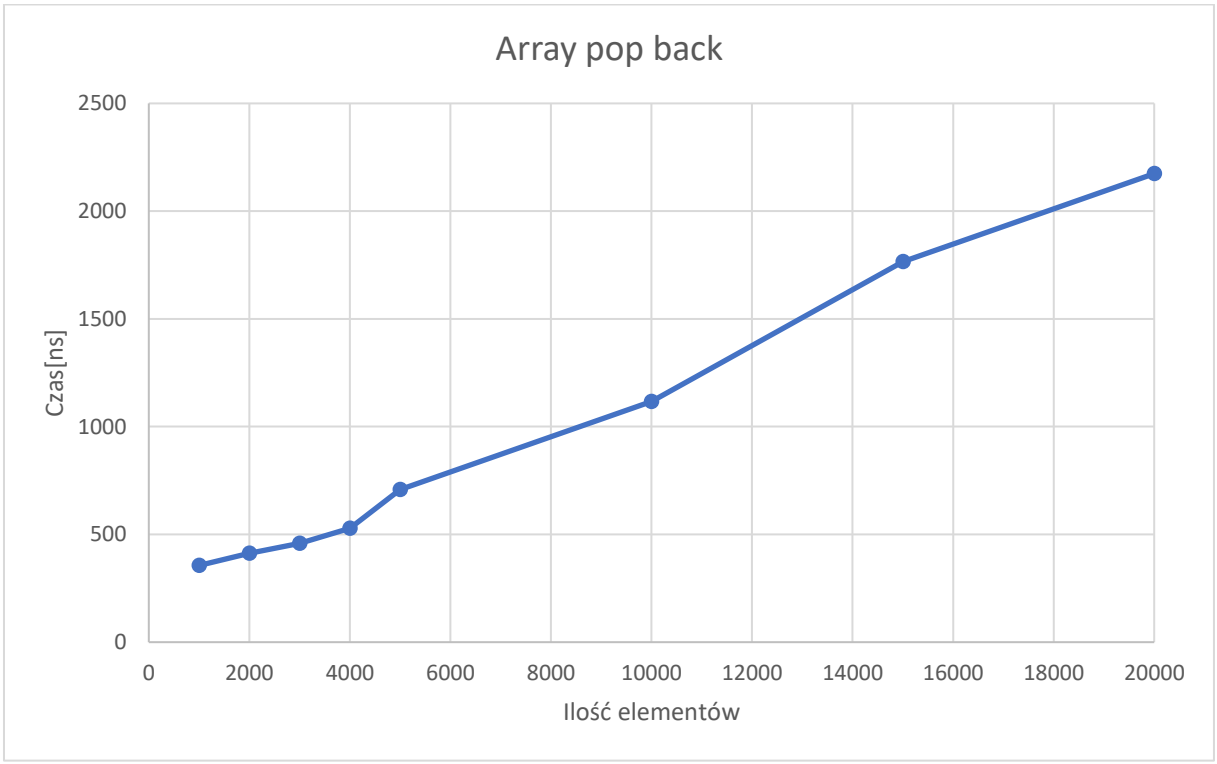
3.Dodanie elementu na przód tablicy



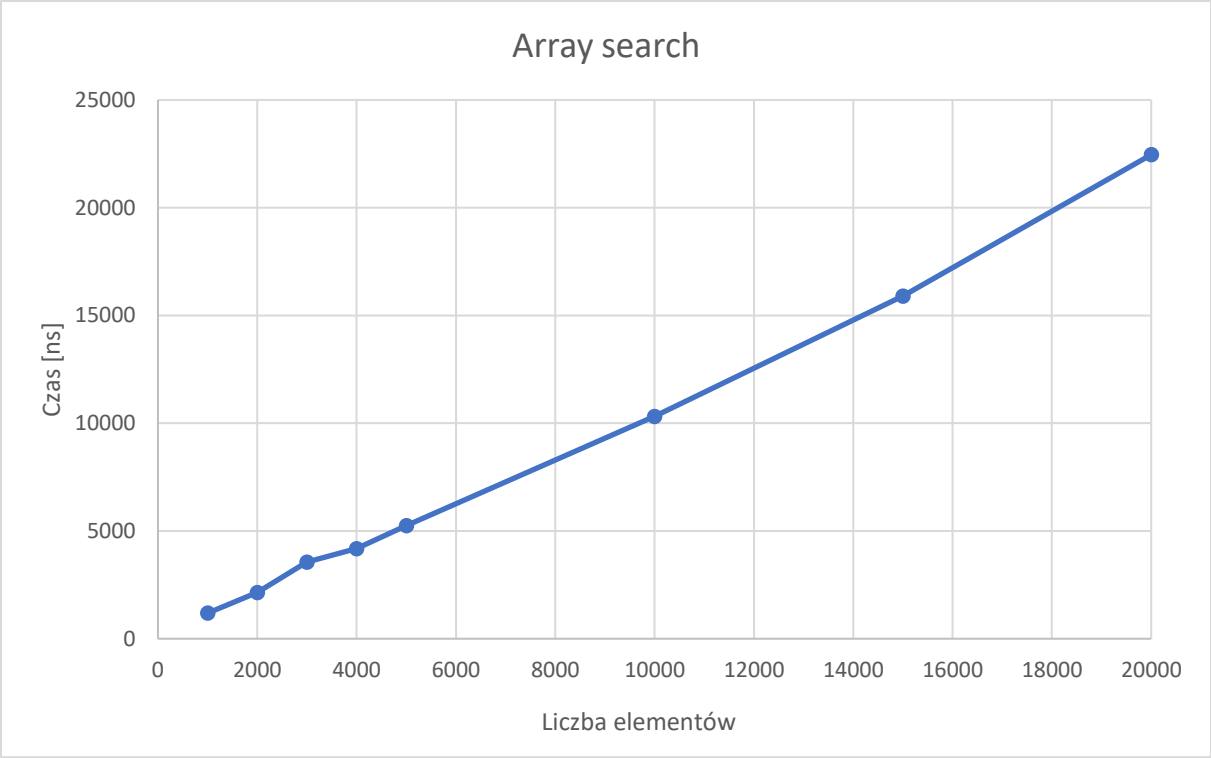
4. Dodanie elementu na tył tablicy



5. Usuniecie elementu z przodu tablicy



6. Usunięcie elementu z tyłu tablicy

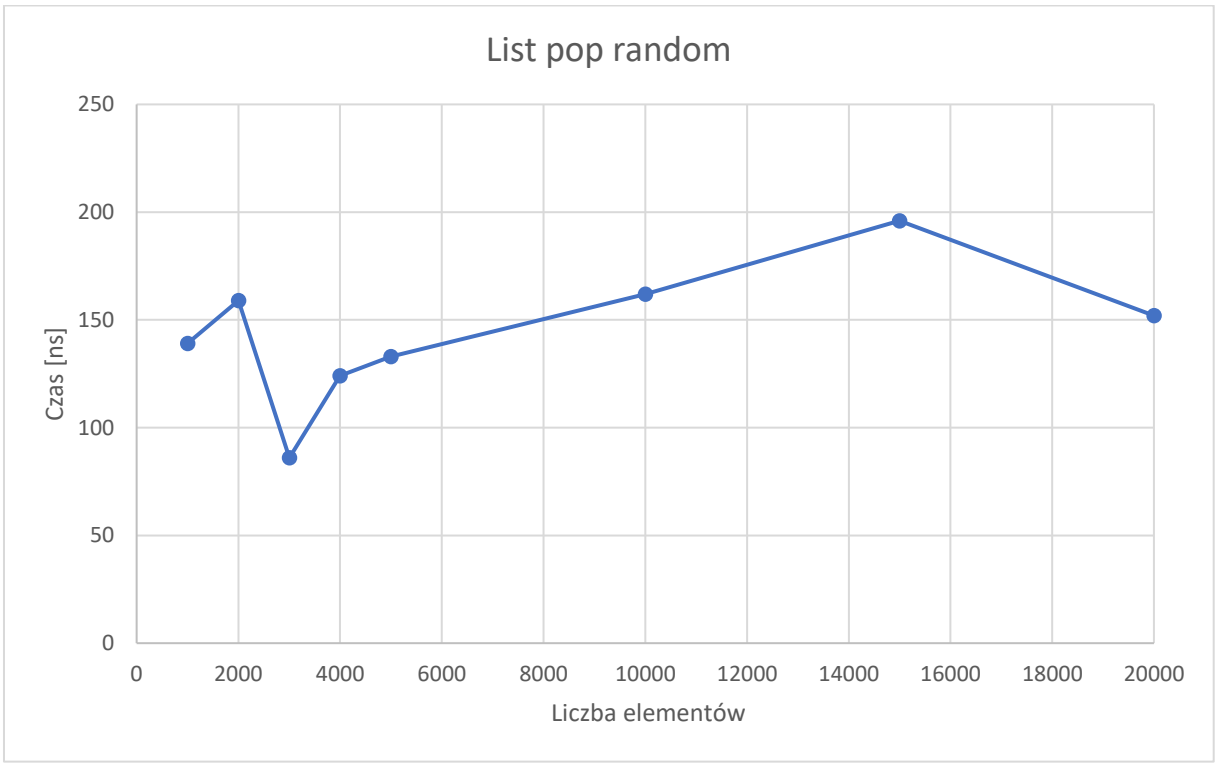


7. Wyszukiwanie losowego elementu w tablicy

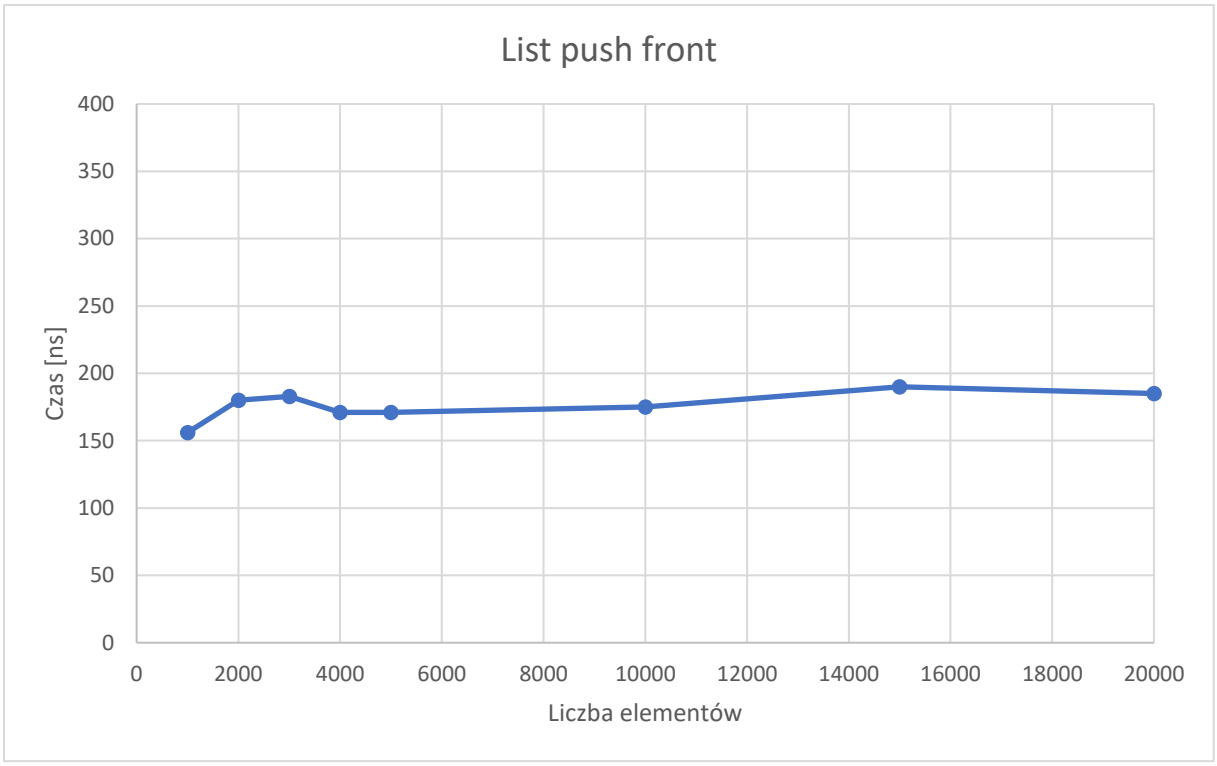
4.2 Lista dwukierunkowa

Liczba Elementów	Push Front	Push Back	Pop front	Pop back	Search	Push	Pop
	[ns]	*[ns]*	*[ns]*	*[ns]*	*[ns]*	*[ns]*	*[ns]*
1000	156	191	172	169	1718	163	139
2000	180	180	187	164	3633	160	159
3000	183	173	183	176	5807	176	86
4000	171	183	165	179	7798	173	124
5000	171	177	175	167	9664	182	133
10000	175	172	167	164	19588	199	162
15000	190	177	176	177	31094	211	196
20000	185	189	173	174	42475	225	152

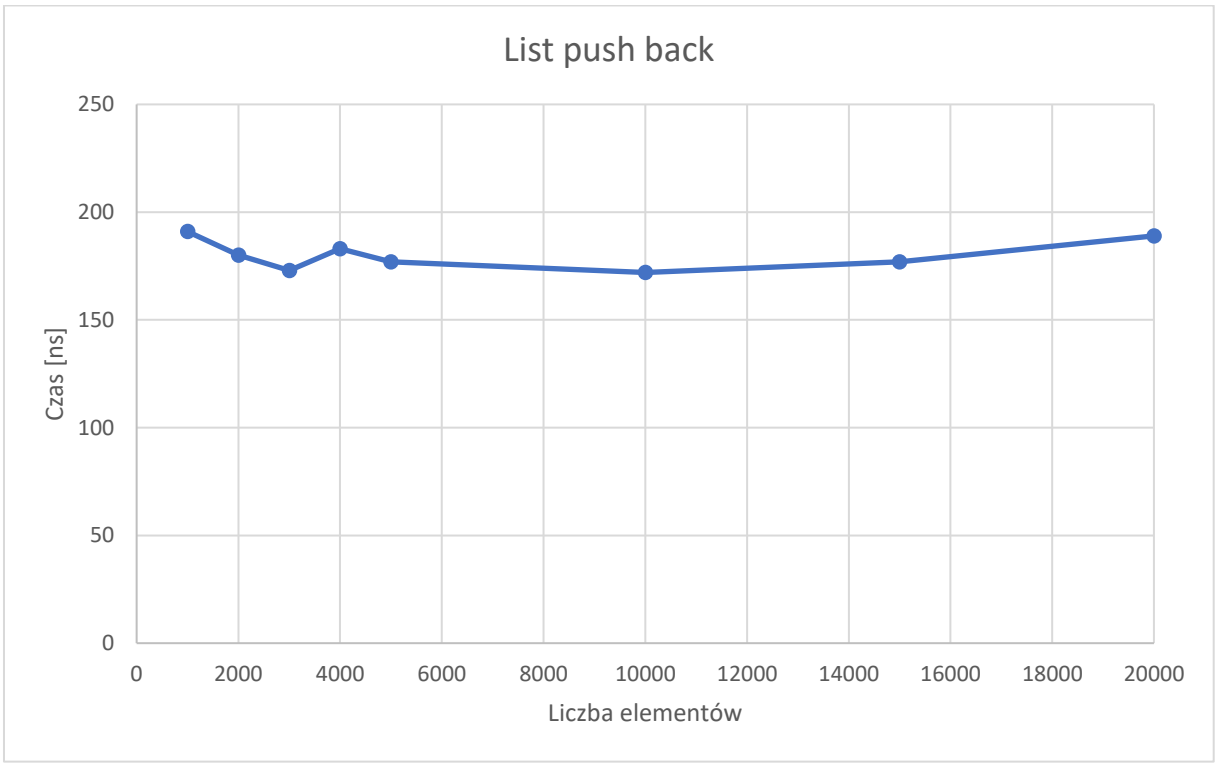
8. Dodanie elementu w losowe miejsce tablicy



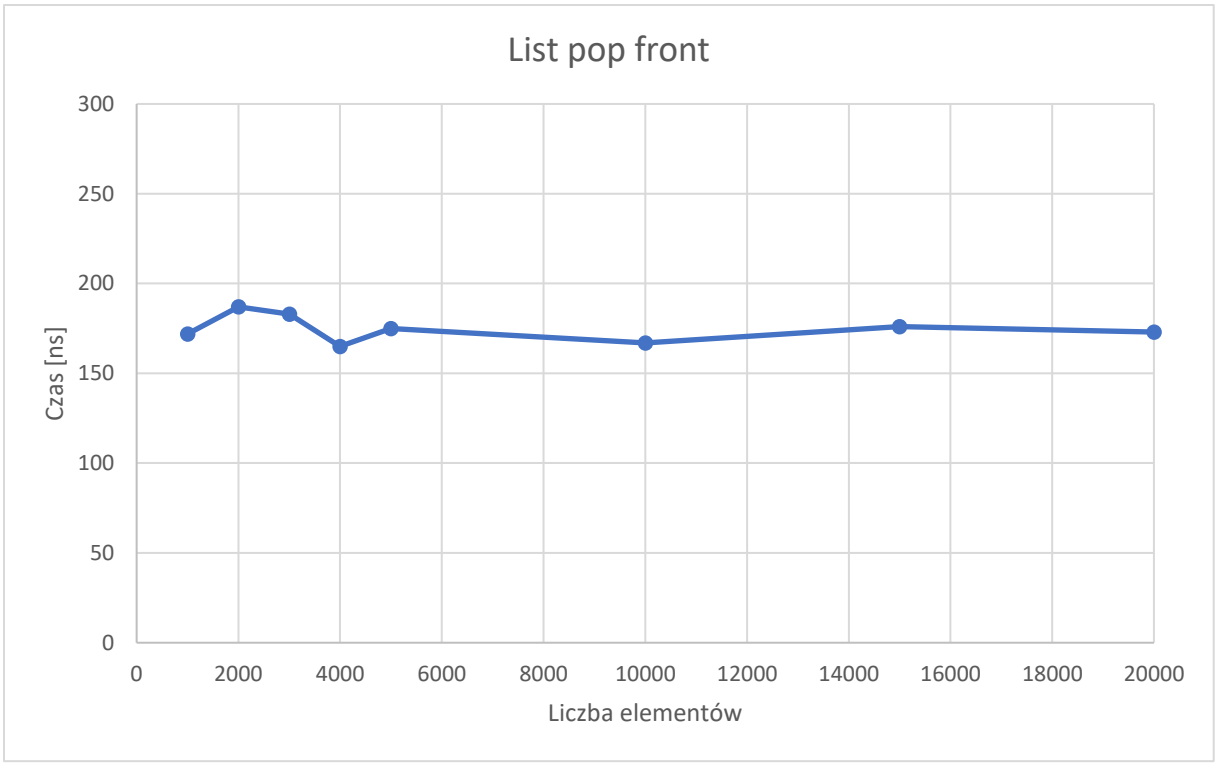
9. Usunięcie losowego elementu z listy



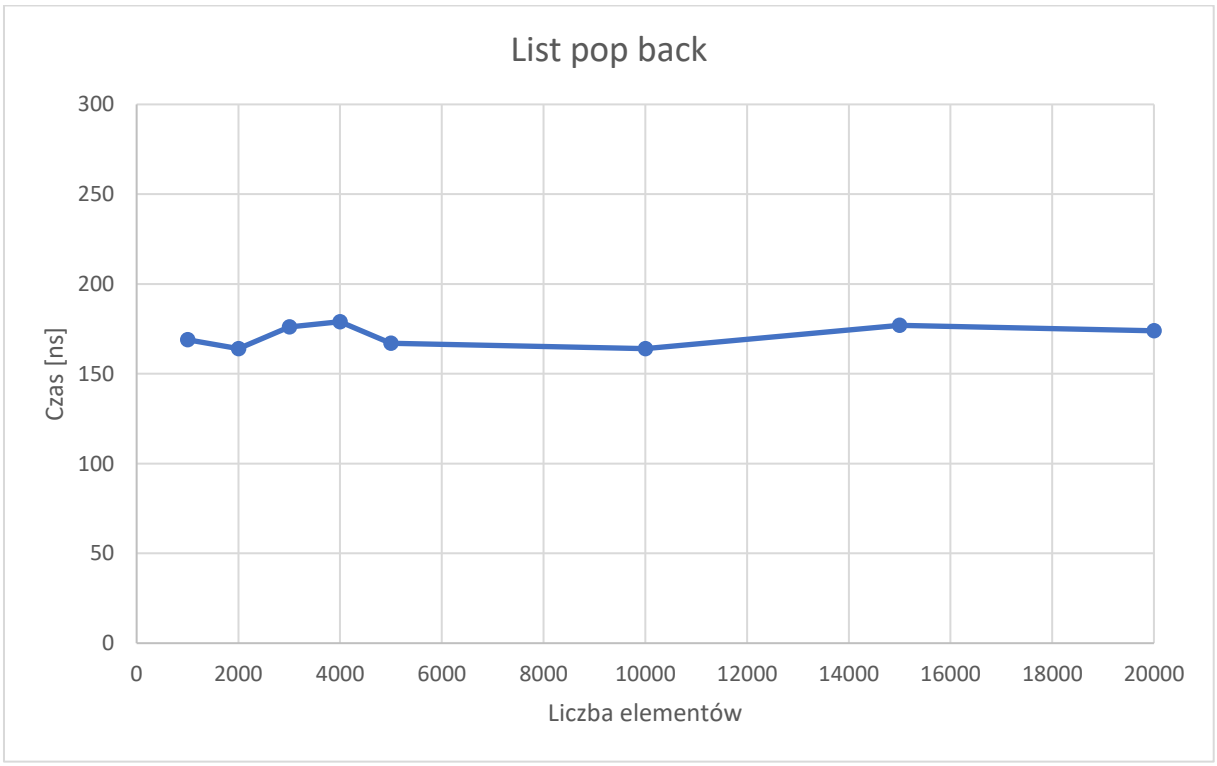
10. Dodanie elementu na początek listy



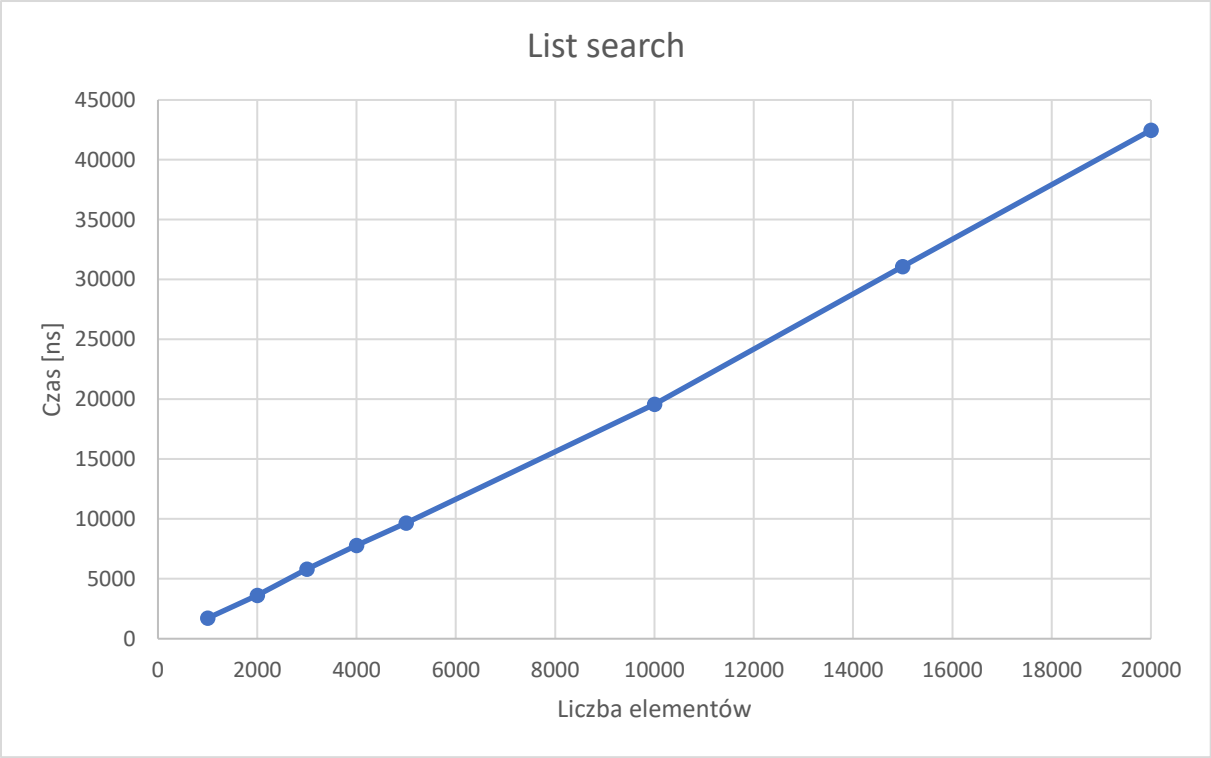
11. Dodanie elementu na koniec listy



12. Usunięcie elementu z początku listy



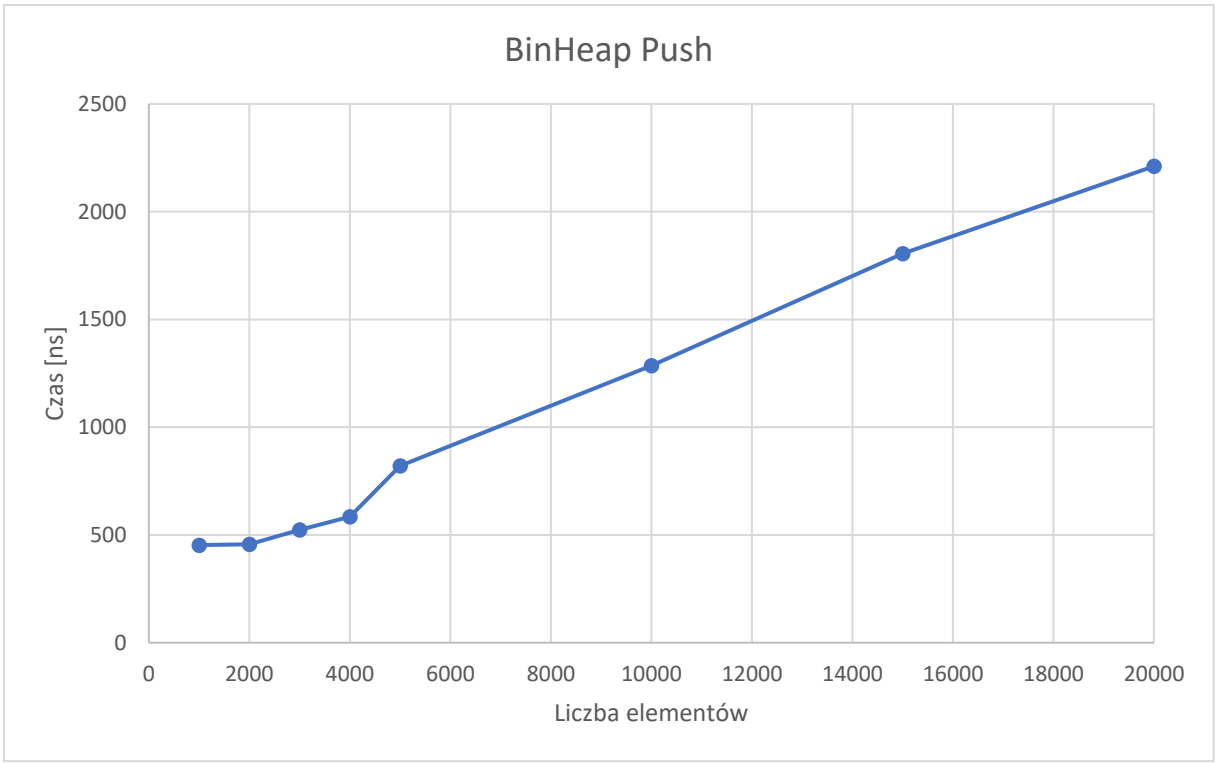
13. Usunięcie elementu z końca listy



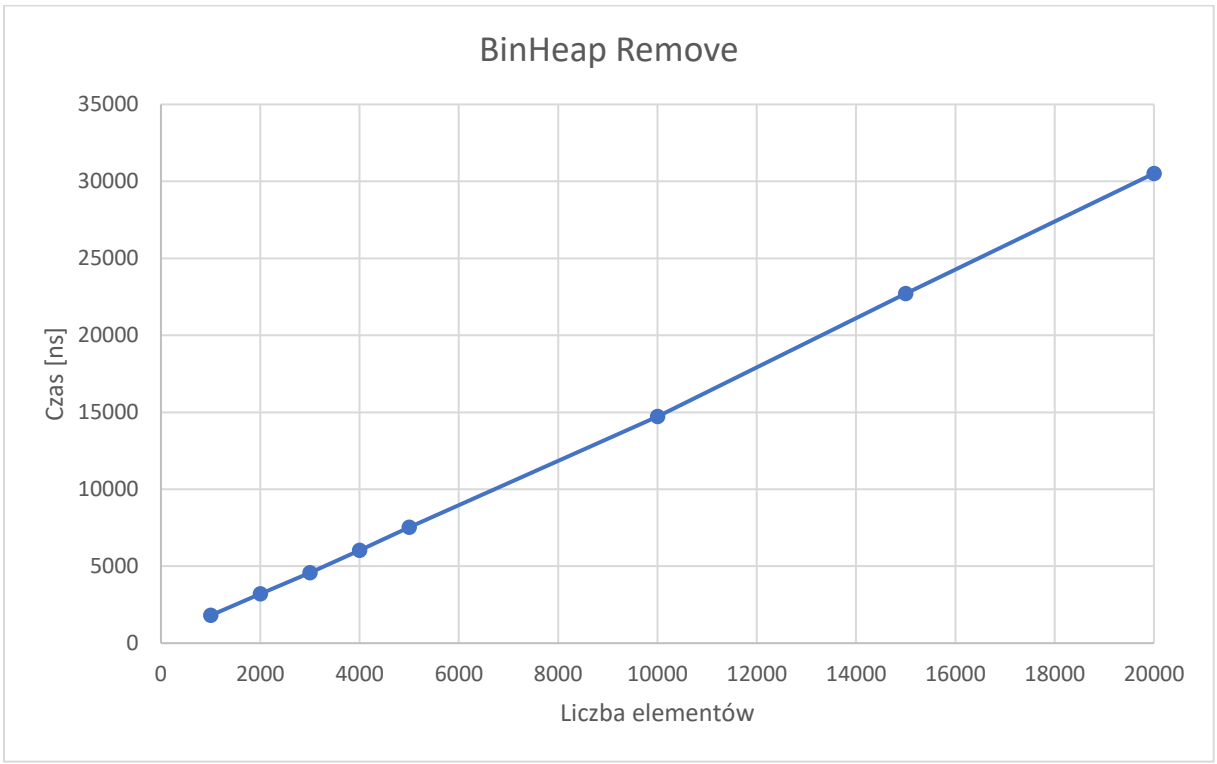
14. Wyszukiwanie elementu w liście

4.3 Kopiec binarny

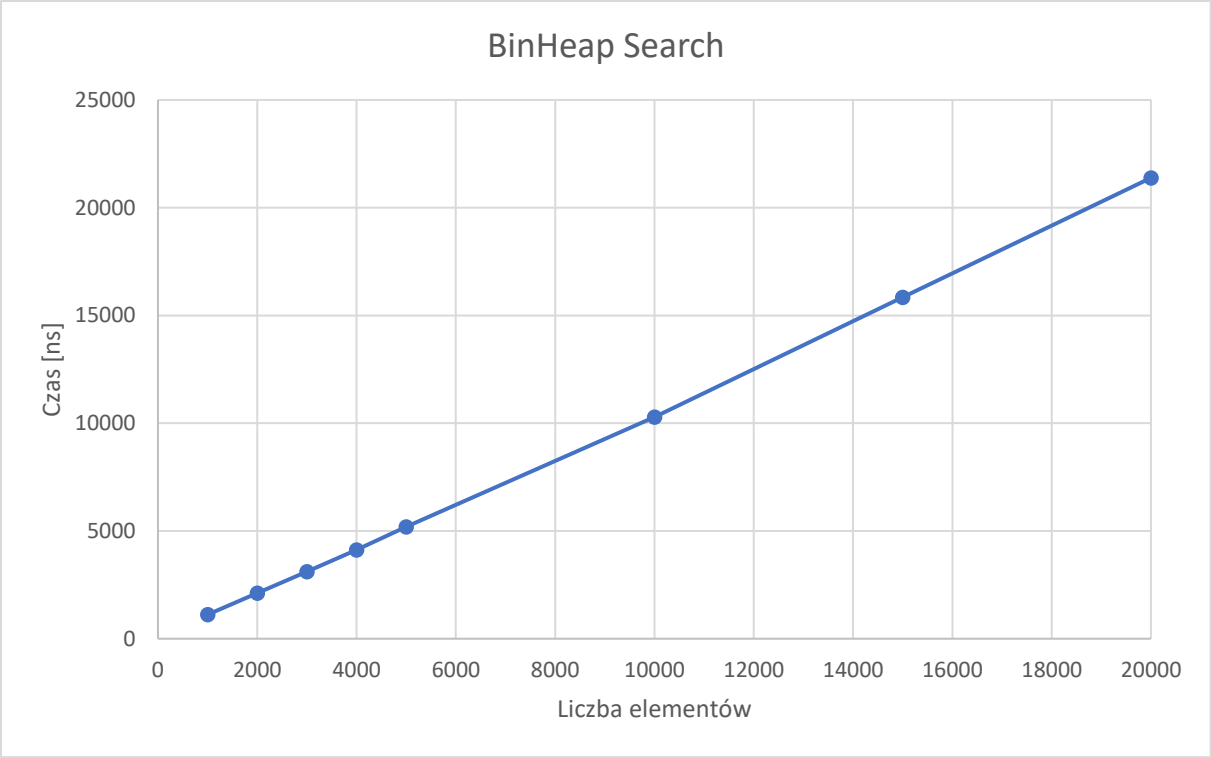
Liczba Elementów	Push	Pop	Search
	[ns]	*[ns]*	*[ns]*
1000	453	1802	1119
2000	457	3200	2119
3000	524	4578	3108
4000	584	6030	4124
5000	821	7521	5194
10000	1286	14720	10290
15000	1805	22706	15838
20000	2211	30513	21383



15. Dodanie elementu do kopca



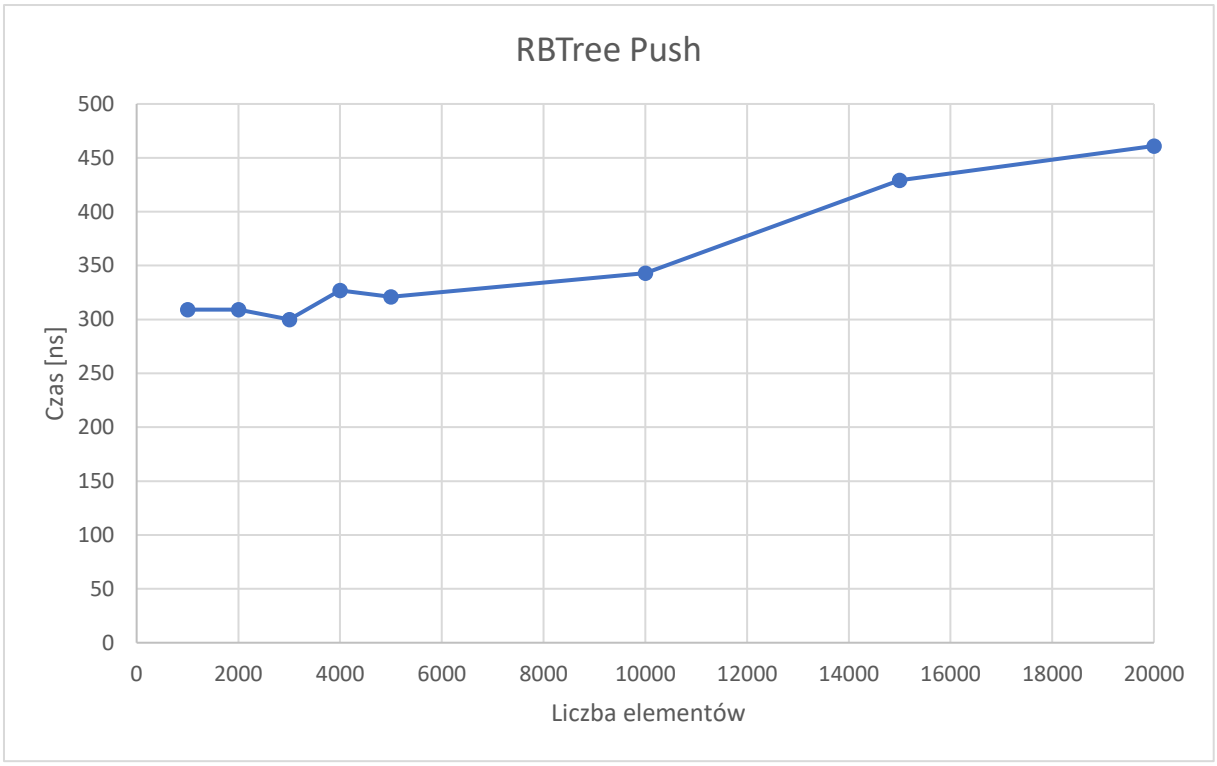
16. Usunięcie elementu z kopca



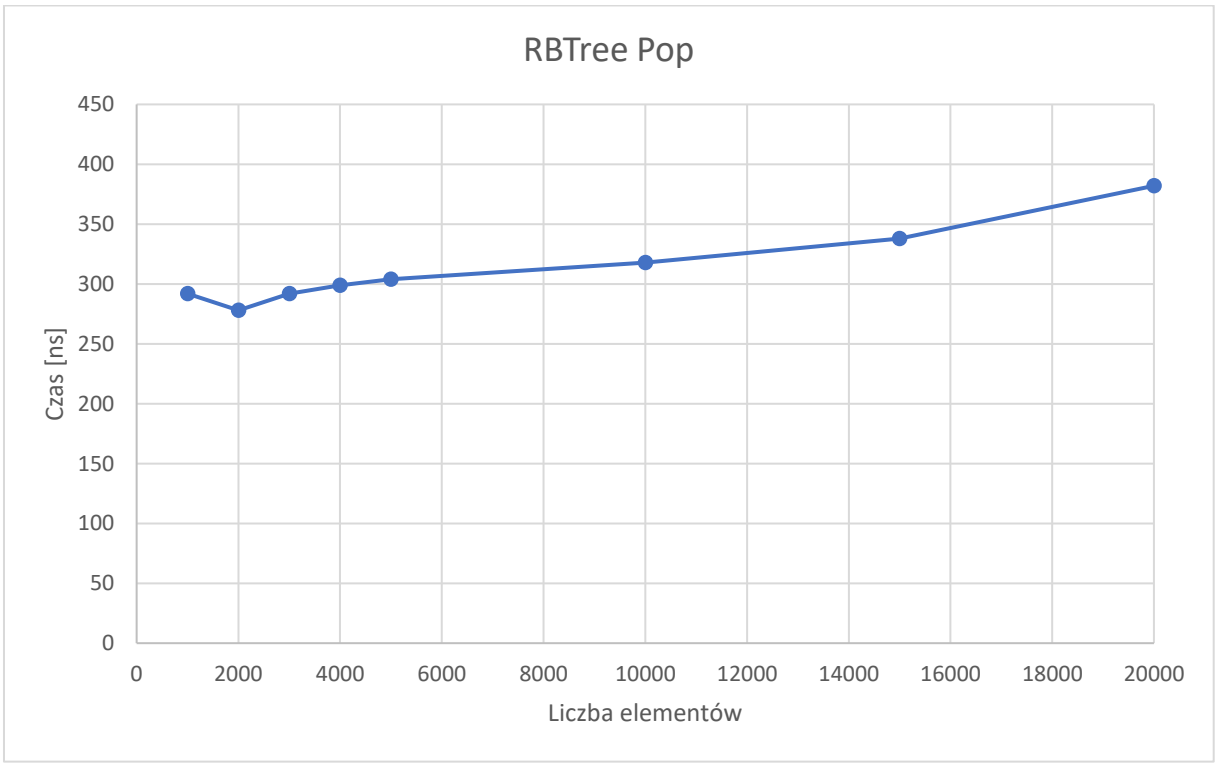
17. Wyszukiwanie elementu w kopcu

4.4 Drzewo czerwono czarne

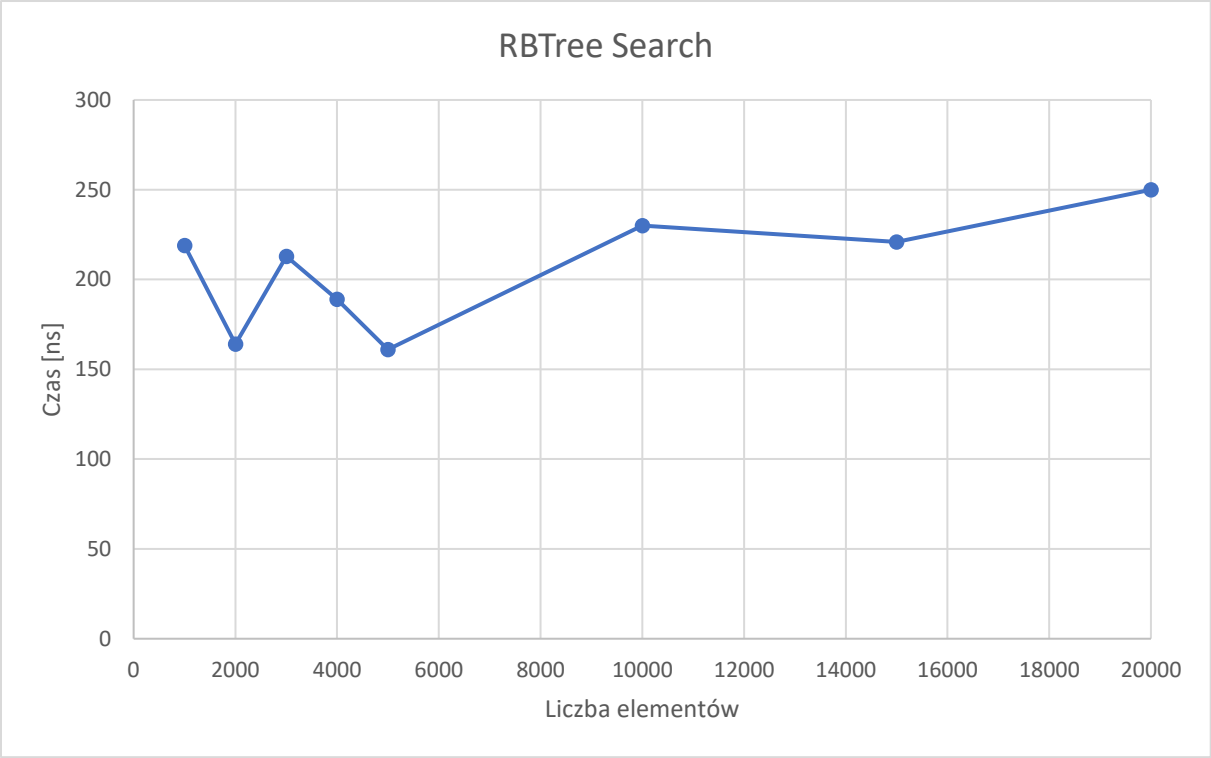
Liczba Elementów	Push	Pop	Search
	[ns]	***[ns]***	***[ns]***
1000	309	292	219
2000	309	278	164
3000	300	292	213
4000	327	299	189
5000	321	304	161
10000	343	318	230
15000	429	338	221
20000	461	382	250



18. Dodanie elementu do drzewa czerwono czarnego



19. Usunięcie elementu z drzewa czerwono czarnego



20. Wyszukiwanie elementu w drzewie czerwono czarnym

5. Wnioski

Czas wykonania operacji na tablicach dynamicznych zmienia się liniowo. Długość wykonywania operacji dodawania, usuwania i wyszukiwania zależy głównie od długości tej struktury.

Operacje dodawania i usuwania elementu na początku i końcu mają praktycznie zerowy czas wykonania. Co do wstawiania i usuwania losowego mogę mieć przypuszczenia, że czas będzie zwiększał się wraz ze wzrostem rozmiaru listy. Operacja wyszukania jest zależna od wielkości struktury.

Wszystkie operacje na kopcu wykazały losową złożoność obliczeniową. Dzieje się tak najprawdopodobniej dlatego, że algorytm mój opierałem na swojej implementacji struktury tablicy, a mało sprawny algorytm dołożył swoje opóźnienia.

Najbardziej efektywną strukturą okazały się drzewa czerwono czarne. W teorii ich złożoność jest logarytmiczna, jednak możemy spokojnie założyć, że czas trwania pomiarów i ilość badanych długości struktury nie była ani trochę reprezentatywna. Na wykresach nadal jednak można zaobserwować lekki wzrost wraz ze zwiększeniem się liczebności elementów.