SPRAWOZDANIE

STRUKTURY DANYCH I ZŁOŻONOŚĆ OBLICZENIOWA

Zadanie projektowe nr 1

BADANIE EFEKTYWNOŚCI OPERACJI NA DANYCH W PODSTAWOWYCH STRUKTURACH DANYCH

Piątek TP, 13:15

Autor:

Patryk Dulęba

1 Wstęp teoretyczny

1.1 Cel

Zadanie projektowe polegało na zaimplementowaniu, oraz dokonaniu pomiaru czasu wykonywania operacji dodawania, usuwania oraz wyszukiwania elementu w poniższych strukturach danych :

- Tablica dynamiczna
- Lista dwukierunkowa
- Kopiec binarny (typu maksimum)
- Drzewo czerwono-czarne

1.2 Założenia

Przed implementacją zostały podane założenia co do sposobu wykonywania zadania:

- Podstawowym elementem struktur jest 4 bajtowa liczba całkowita (**int**)
- Wszystkie zaimplementowane struktury danych są alokowane dynamicznie.
- W przypadku tablicy oraz listy należy rozpatrzeć osobno przypadki dodawania oraz usuwania na początku, na końcu oraz ze środka struktury.
- Program został napisany w języku **C++** przy użyciu środowiska **Visual Studio**

1.3 Złożoność obliczeniowa

Badane operacje wykonywane na zadanych strukturach posiadają tzw. książkową złożoność obliczeniową algorytmu, czyli według definicji ilość zasobów komputerowych potrzebnych do jego wykonania. Złożoność możemy podzielić na trzy typy:

- Złożoność optymistyczna – najkrótszy możliwy czas wykonania algorytmu.
- Złożoność średnia – standardowy czas wykonywania algorytmu.
- Złożoność pesymistyczna – najdłuższy możliwy czas wykonania algorytmu.

Poniżej w tabelach przedstawione zostały książkowe złożoności obliczeniowe badanych struktur danych:

Tablica Dynamiczna

	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
Średnia Złożoność	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
Pesymistyczna Złożoność	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)

Lista dwukierunkowa

	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
Średnia Złożoność	O(1)	O(1)	O(n)	O(1)	O(1)	O(n)	O(n)
Pesymistyczna Złożoność	O(1)	O(1)	O(n)	O(1)	O(1)	O(n)	O(n)

Kopiec binarny

	Dodawanie	Usuwanie	Wyszukiwanie
Średnia Złożoność	O(1)	$O(\log_2(n))$	O(n)
Pesymistyczna Złożoność	$O(\log_2(n))$	$O(\log_2(n))$	O(n)

Drzewo czerwono-czarne

	Dodawanie	Usuwanie	Wyszukiwanie
Średnia Złożoność	$O(\log_2(n))$	$O(\log_2(n))$	$O(\log_2(n))$
Pesymistyczna Złożoność	$O(\log_2(n))$	$O(\log_2(n))$	$O(\log_2(n))$

2 Plan eksperymentu

W programie zaimplementowane zostały wszystkie wymagane funkcje tzn. dodawanie, usuwanie oraz wyszukiwanie elementu. Zgodnie z założeniami w tablicy i liście możemy dodawać oraz usuwać elementy na początku w środku oraz na końcu. Dodatkowo każda struktura posiada funkcję wczytania danych z pliku, wygenerowania danych automatycznie ze wskazaną ilością oraz zakresem i testowania struktury.

Do przeprowadzenia eksperymentu użyto wielkości struktur: **100, 200, 500, 1000, 2000, 5000, 10000, 15000, 20000**.

Pomiary dla wszystkich struktur z wyjątkiem drzewa czerwono-czarnego przeprowadzone zostały dla dwóch przedziałów wartości elementów: **[1;100]** oraz **[1;max_int],** natomiast dla drzewa czerwono-czarnego był to jeden przedział **[1;20000].** Powodem tego jest specyfika budowy drzewa, gdzie elementy nie mogą się powtarzać.

Pomiar czasu poszczególnych operacji został wykonany za pomocą: **QueryPerformanceCounter.**

Przebieg pomiaru wyglądał w następujący sposób:

1) Podanie wielkości struktury oraz zakres wartości jej elementów
2) Wygenerowanie i wypełnienie struktury danymi na pomocą funkcji **rand()**
3) Wykonanie danej funkcji wraz z pomiarem czasu
4) Dodanie uzyskanego pomiaru czasu do zmiennej liczącej

Po 100 krotnym wykonaniu powyższych operacji funkcja liczyła i wypisywała średnią

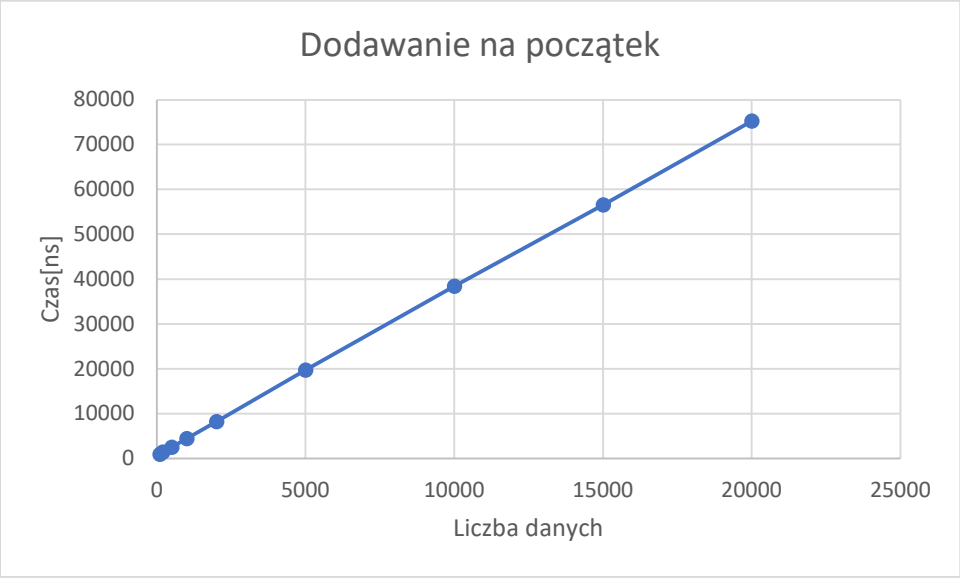
3 Zestawienie wyników

3.1 Tablica dynamiczna

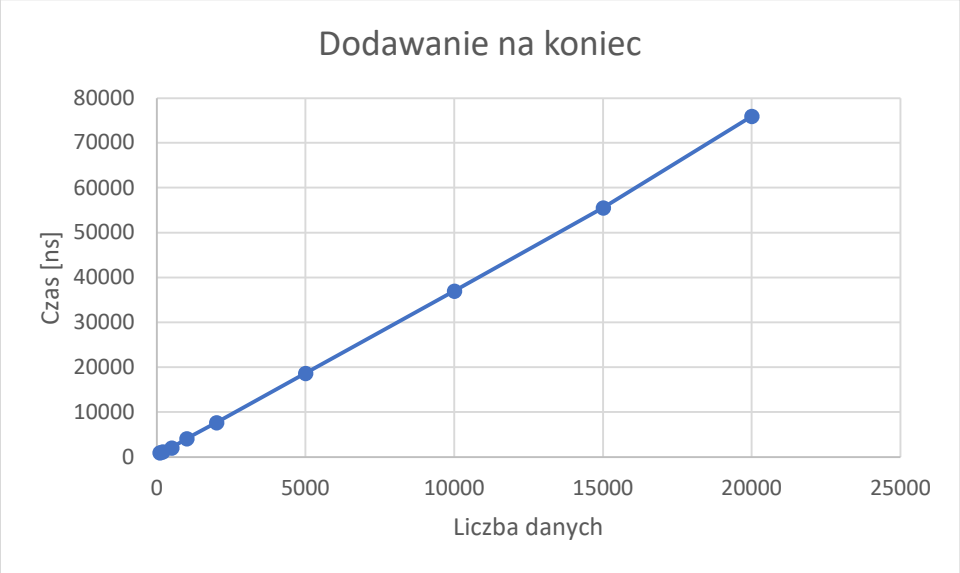
Tabela 1: Pomiary dla tablicy dynamicznej wypełnionej wartościami z przedziału [1;100]

	Liczba elementów	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]	[ns]	[ns]	[ns]	[ns]
1	100	929	966	996	943	949	935	387
2	200	1436	1138	1184	1209	1221	1197	610
3	500	2464	2039	2241	2263	2401	2255	1313
4	1000	4426	4072	4203	4133	4319	4018	2559
5	2000	8176	7666	8144	7958	8221	7955	5127
6	5000	19677	18612	19576	19933	20195	19497	12956
7	10000	38376	36978	37961	38195	38973	37794	25116
8	15000	56493	55516	56028	56806	58139	55998	37406
9	20000	75223	75959	74312	75766	76328	74550	49864

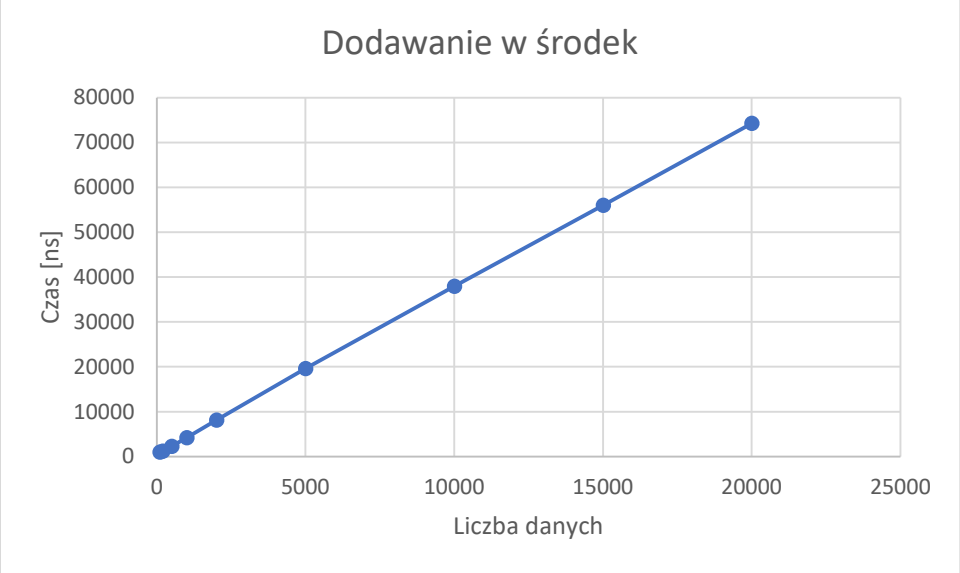
Wykres 1: Dodawanie elementu na początek tablicy wypełnionej wartościami [1;100]



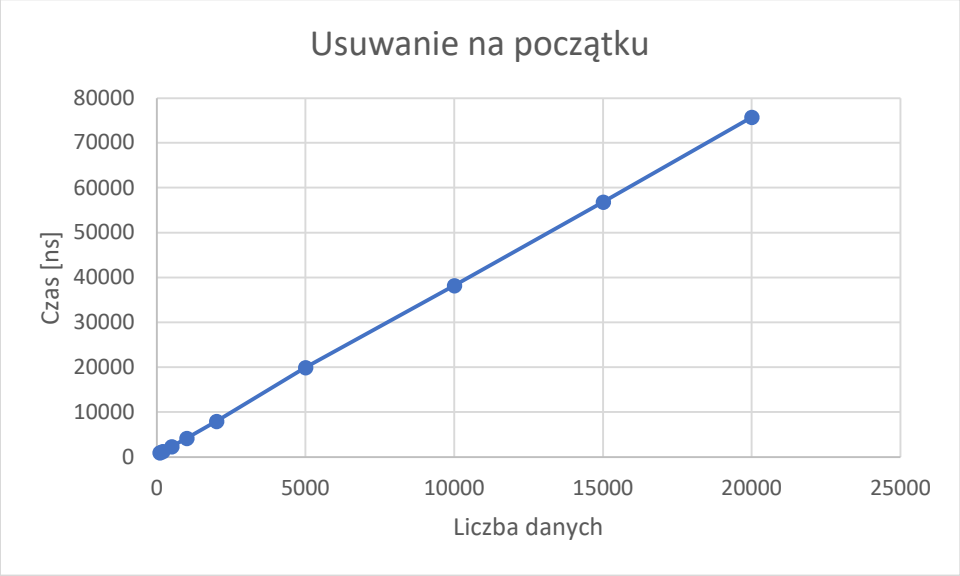
Wykres 2: Dodawanie elementu na koniec tablicy wypełnionej wartościami [1;100]



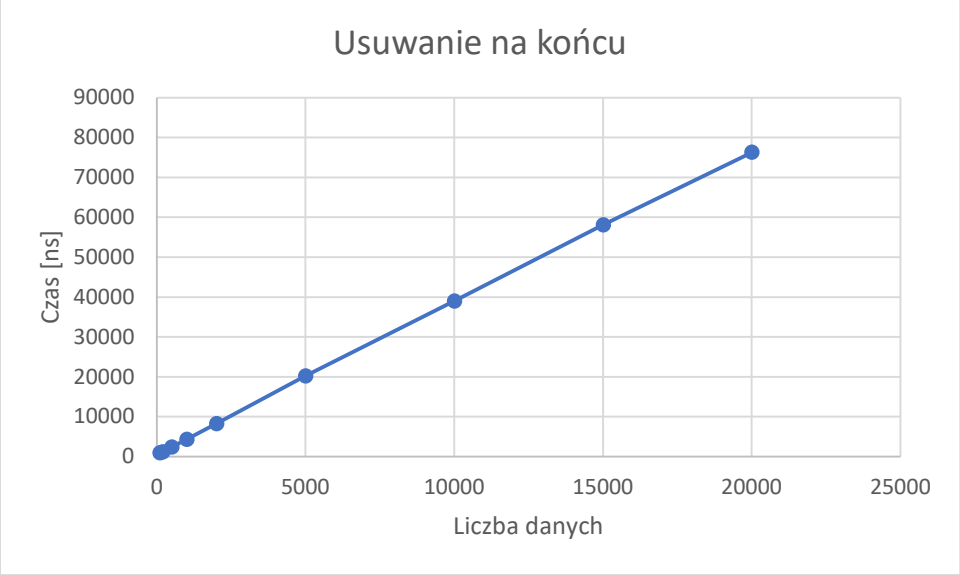
Wykres 3: Dodawanie elementu w środek tablicy wypełnionej wartościami [1;100]



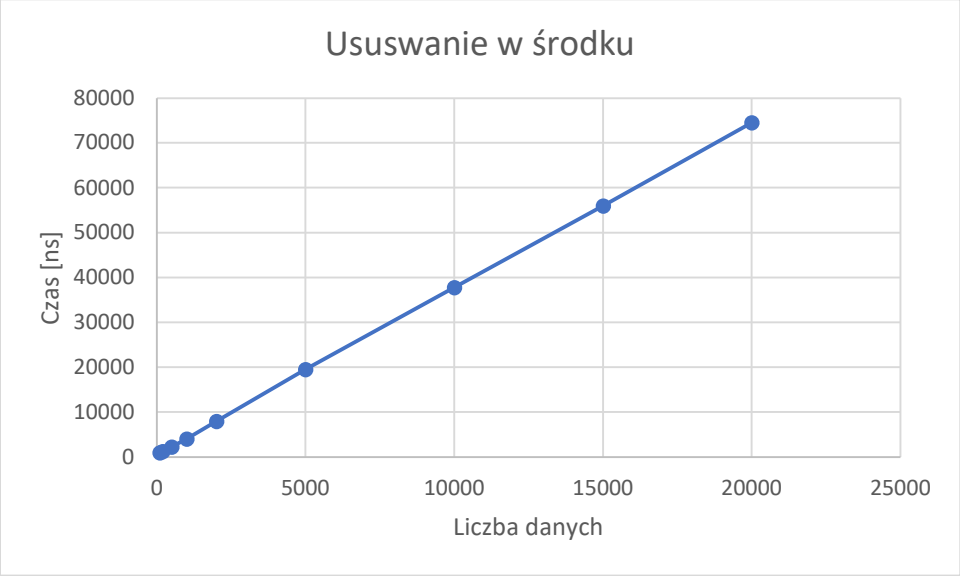
Wykres 4: Usuwanie elementu na początku tablicy wypełnionej wartościami [1;100]



Wykres 5: Usuwanie elementu na końcu tablicy wypełnionej wartościami [1;100]



Wykres 6: Usuwanie elementu w środku tablicy wypełnionej wartościami [1;100]



Wykres 7: Wyszukiwanie elementu w tablicy wypełnionej wartościami [1;100]

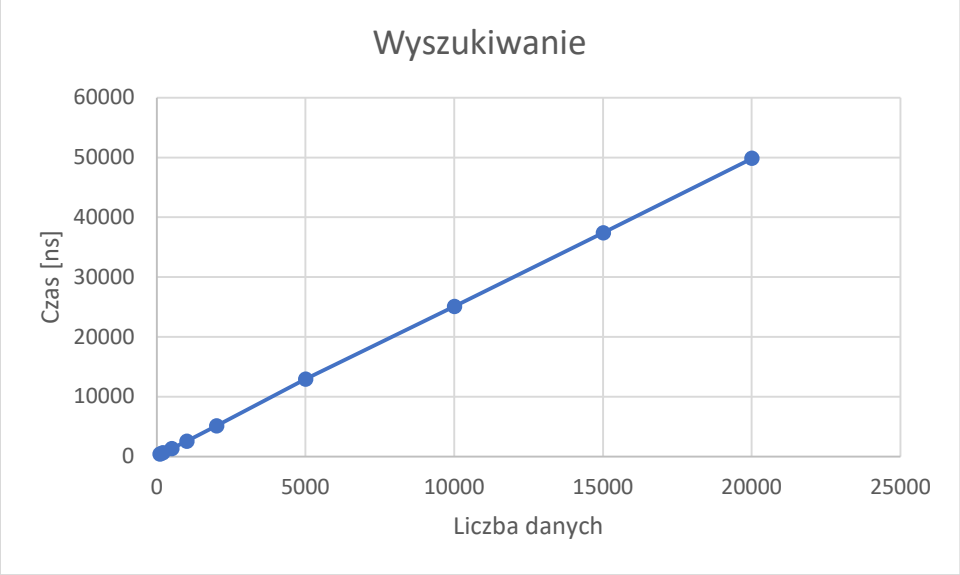
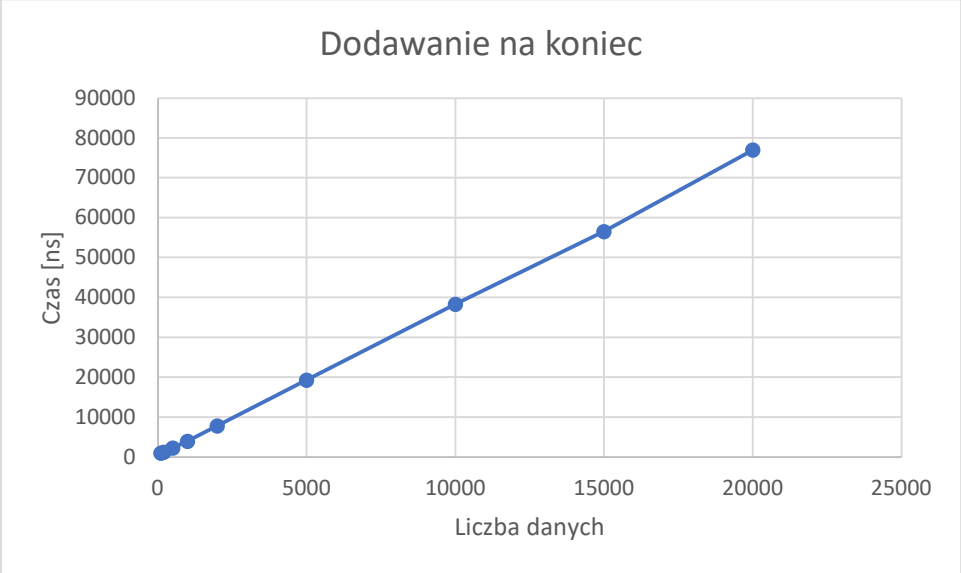


Tabela 2: Pomiary dla tablicy dynamicznej wypełnionej wartościami z przedziału [1;max_int]

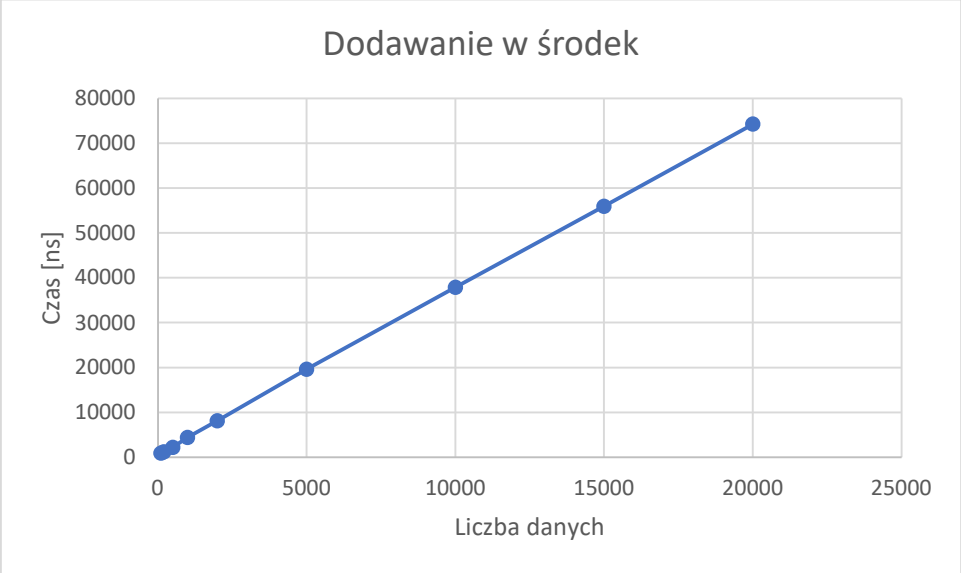
	Liczba elementów	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]	[ns]	[ns]	[ns]	[ns]
1	100	968	971	941	760	974	943	388
2	200	1224	1217	1197	1205	1221	1168	596
3	500	2309	2234	2229	2374	2291	2278	1318
4	1000	4175	3937	4407	4189	4293	4283	2576
5	2000	8231	7762	8101	8251	8129	8152	5229
6	5000	20047	19284	19608	19872	20266	19430	12974
7	10000	38332	38347	37891	38099	38867	37899	25085
8	15000	56543	56561	55908	56851	57900	55899	37477
9	20000	75042	76982	74258	75161	76459	74436	49583

Wykres 8: Dodawanie elementu na początek tablicy wypełnionej wartościami [1;max_int]

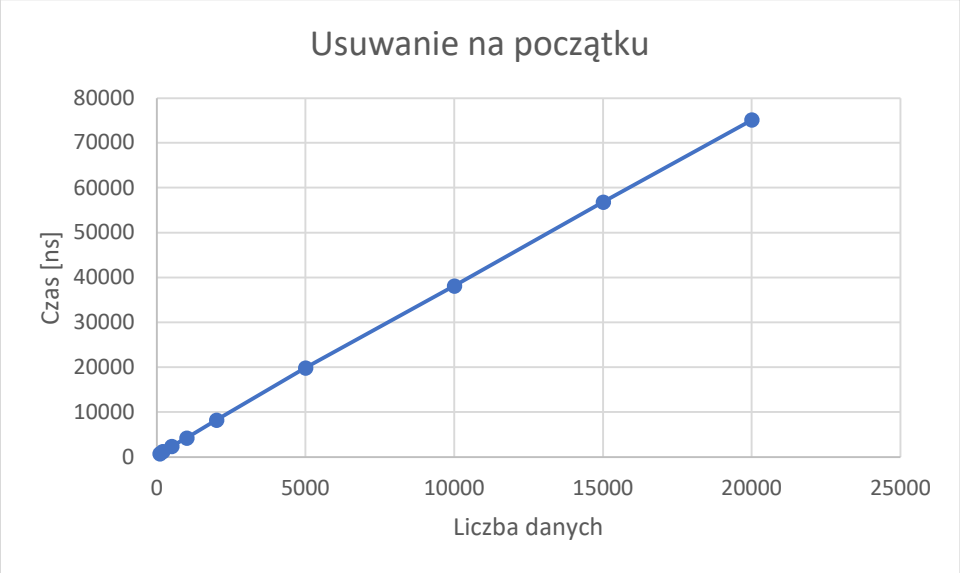
Wykres 9: Dodawanie elementu na koniec tablicy wypełnionej wartościami [1;max_int]



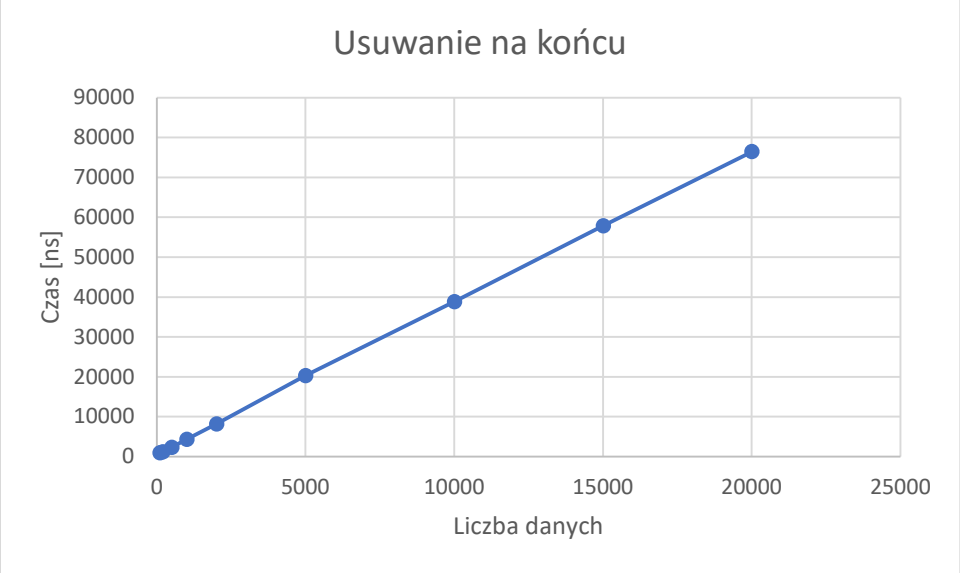
Wykres 10: Dodawanie elementu w środek tablicy wypełnionej wartościami [1;max_int]



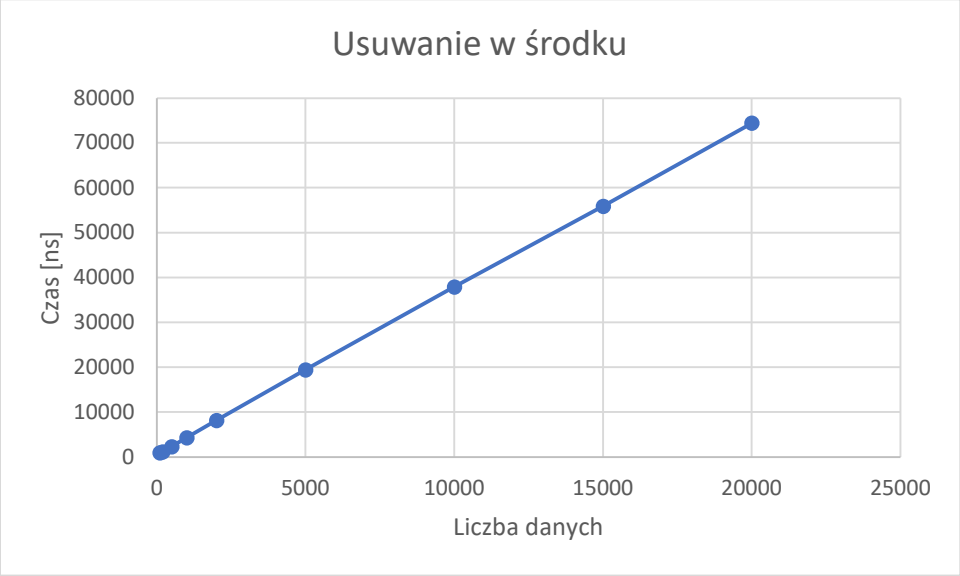
Wykres 11: Usuwanie elementu na początku tablicy wypełnionej wartościami [1;max_int]



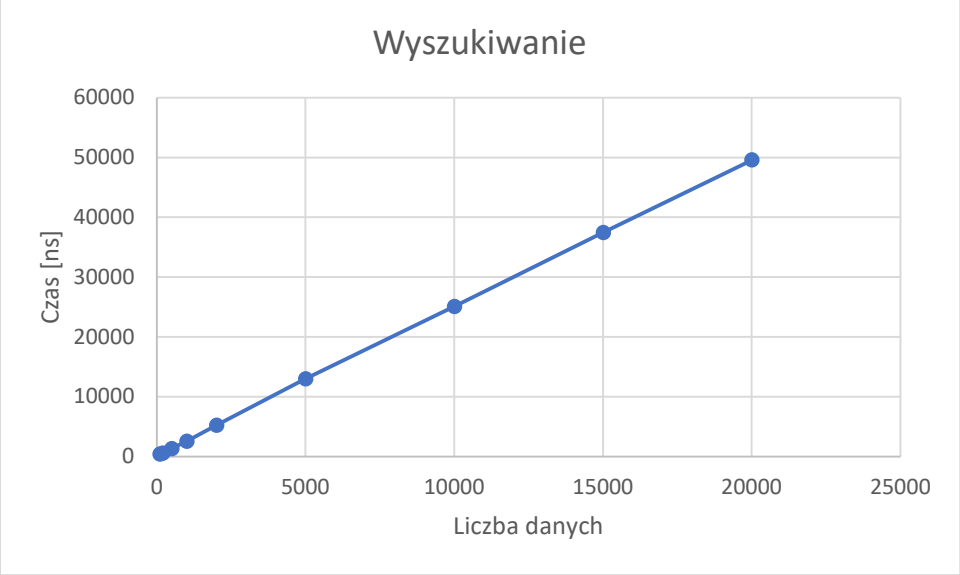
Wykres 12: Usuwanie elementu na końcu tablicy wypełnionej wartościami [1;max_int]



Wykres 13: Usuwanie elementu w środku tablicy wypełnionej wartościami [1;max_int]



Wykres 14: Wyszukiwanie elementu w tablicy wypełnionej wartościami [1;max_int]

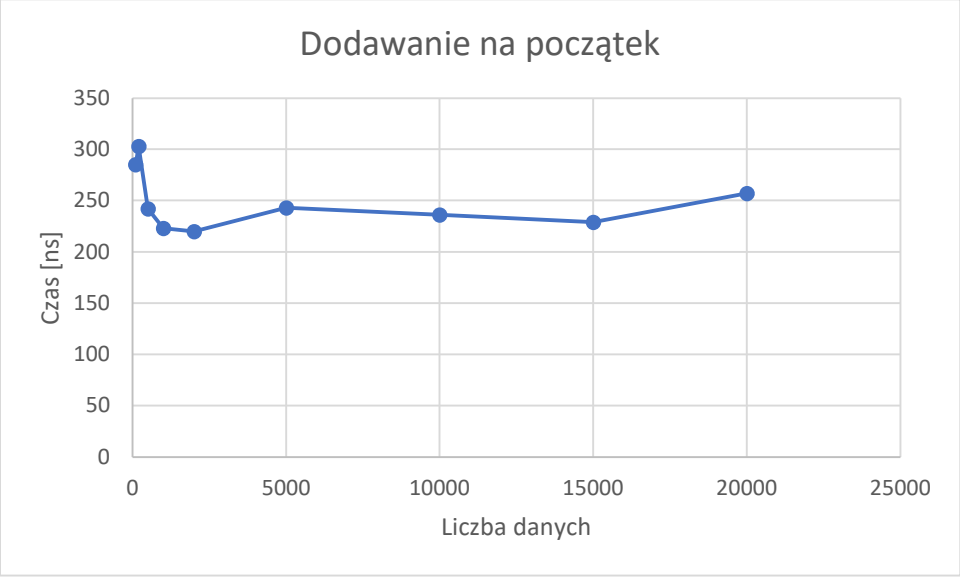


3.2 Lista dwukierunkowa

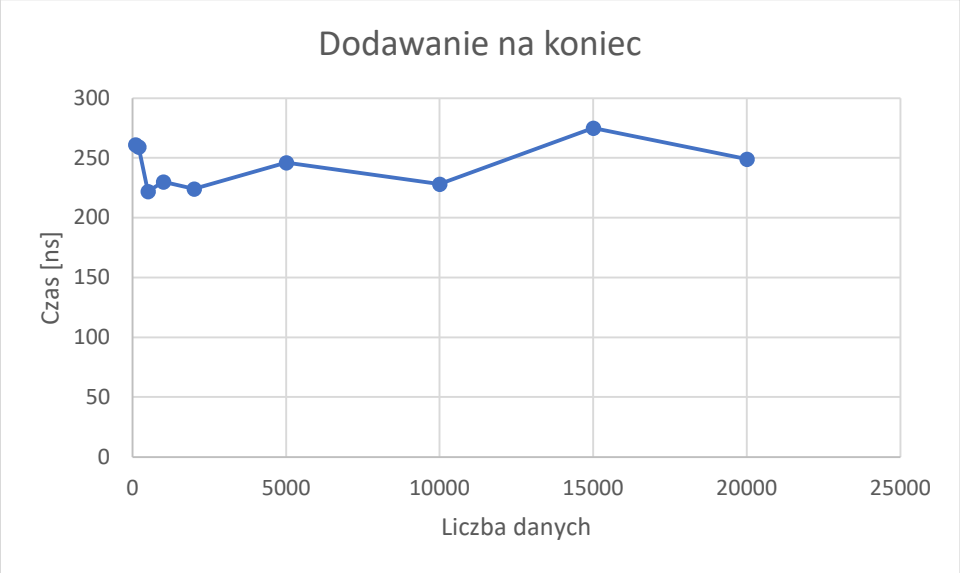
Tabela 3: Pomiary dla listy dwukierunkowej wypełnionej wartościami z przedziału [1;100]

	Liczba elementów	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]	[ns]	[ns]	[ns]	[ns]
1	100	285	261	475	290	309	491	579
2	200	303	259	815	274	285	786	1090
3	500	242	222	1590	249	265	1472	2549
4	1000	223	230	2330	230	251	2321	4397
5	2000	220	224	4466	265	253	4500	8671
6	5000	243	246	12878	284	270	14434	27004
7	10000	236	228	30693	310	288	31633	61091
8	15000	229	275	51631	357	351	53406	98354
9	20000	257	249	79867	390	379	81388	164241

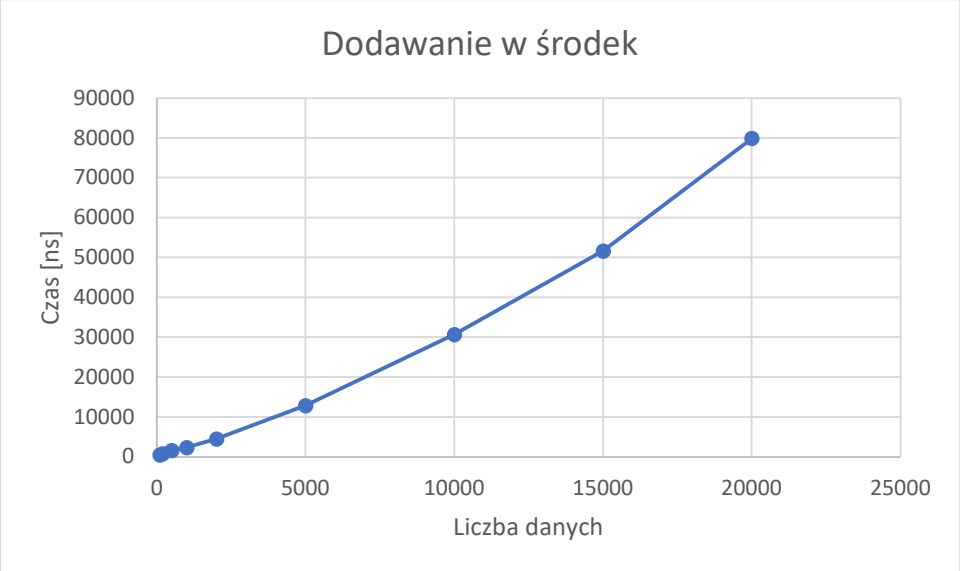
Wykres 15: Dodawanie elementu na początek listy wypełnionej wartościami [1;100]



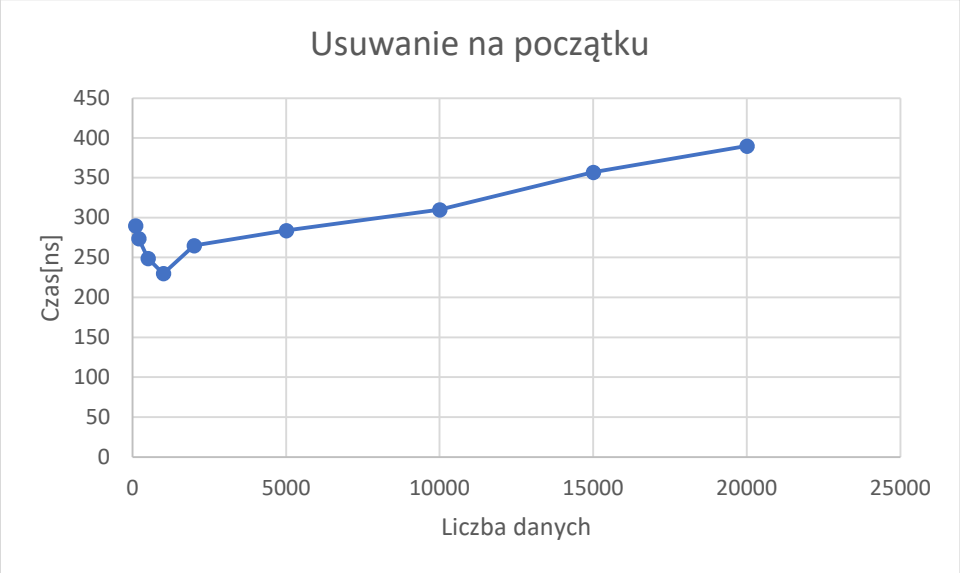
Wykres 16: Dodawanie elementu na koniec listy wypełnionej wartościami [1;100]



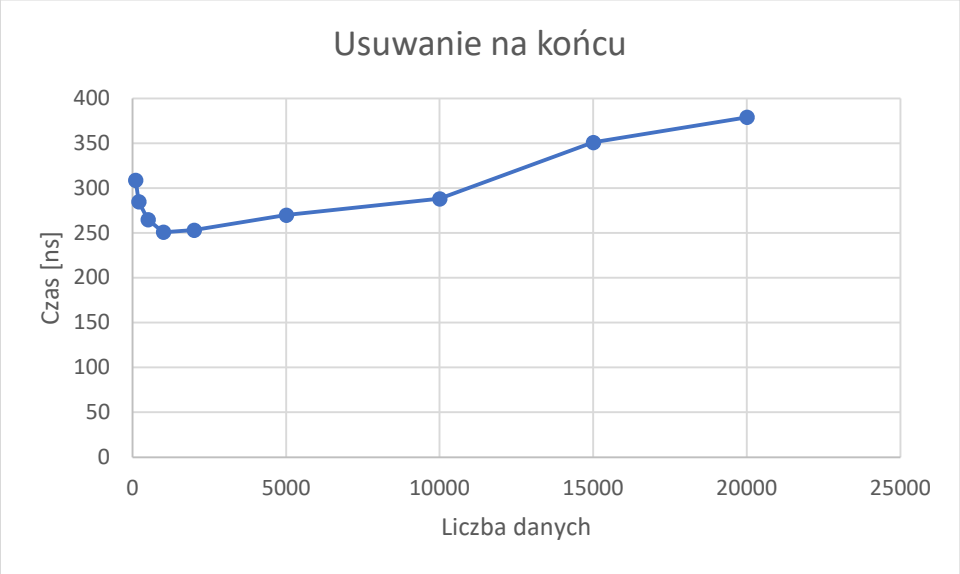
Wykres 17: Dodawanie elementu w środek listy wypełnionej wartościami [1;100]



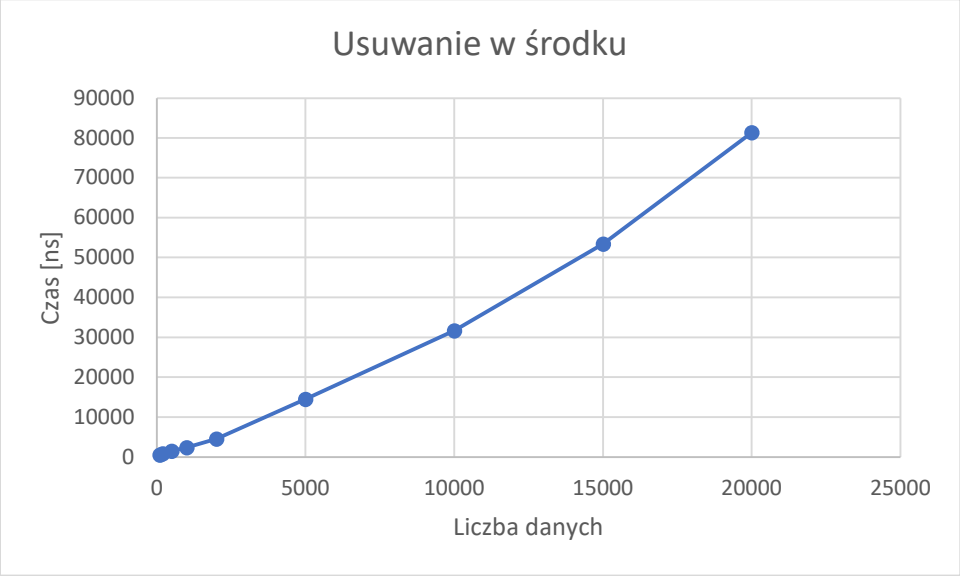
Wykres 18: Usuwanie elementu na początku listy wypełnionej wartościami [1;100]



Wykres 19: Usuwanie elementu na końcu listy wypełnionej wartościami [1;100]



Wykres 20: Usuwanie elementu w środku listy wypełnionej wartościami [1;100]



Wykres 21: Wyszukiwanie elementu w liście wypełnionej wartościami [1;100]

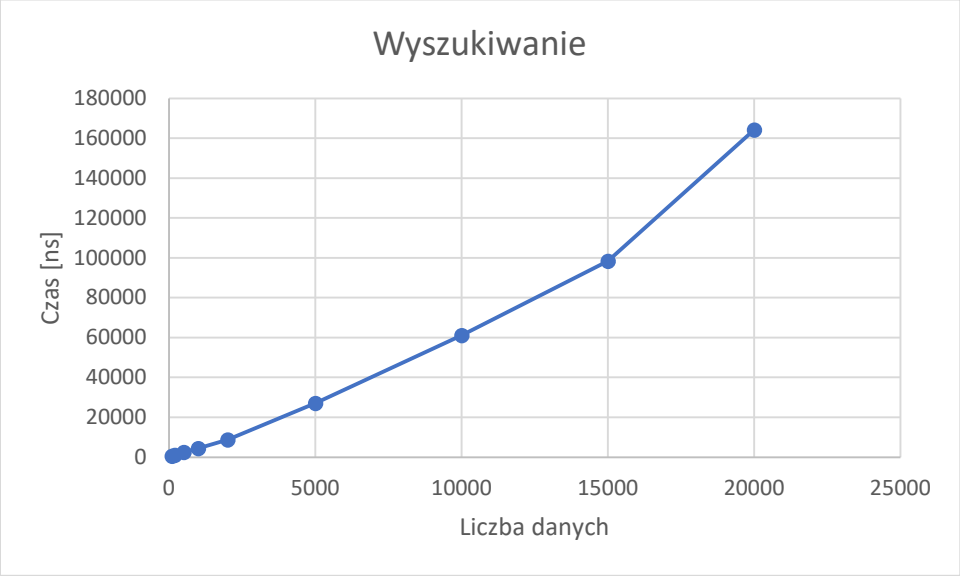
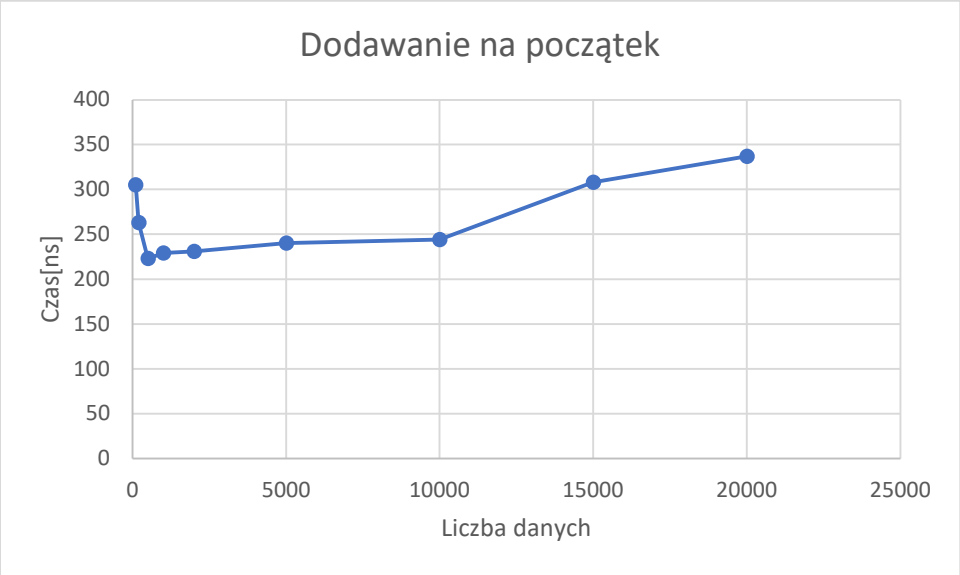


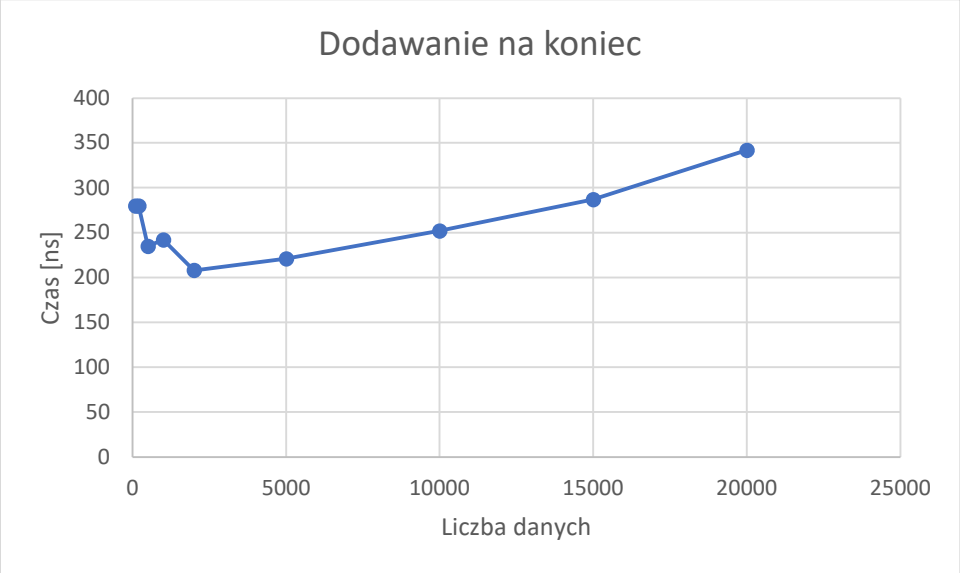
Tabela 4: Pomiary dla listy dwukierunkowej wypełnionej wartościami z przedziału [1;max_int]

	Liczba elementów	Dodawanie na początek	Dodawanie na koniec	Dodawanie w środek	Usuwanie na początku	Usuwanie na końcu	Usuwanie w środku	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]	[ns]	[ns]	[ns]	[ns]
1	100	305	280	531	312	319	544	616
2	200	263	280	779	281	275	830	1044
3	500	223	235	1298	260	275	1357	2421
4	1000	229	242	2350	272	246	2402	4498
5	2000	231	208	4430	292	255	4484	8656
6	5000	240	221	15266	334	286	12383	24712
7	10000	244	252	30782	342	301	30517	61541
8	15000	308	287	46193	334	340	46293	96200
9	20000	337	342	70523	397	358	69190	142178

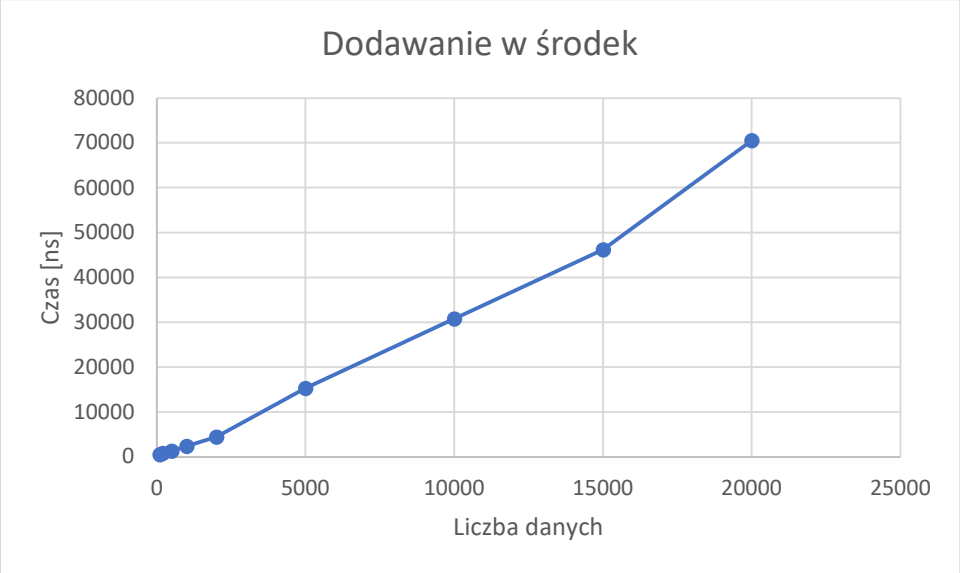
Wykres 22: Dodawanie elementu na początek listy wypełnionej wartościami [1;max_int]



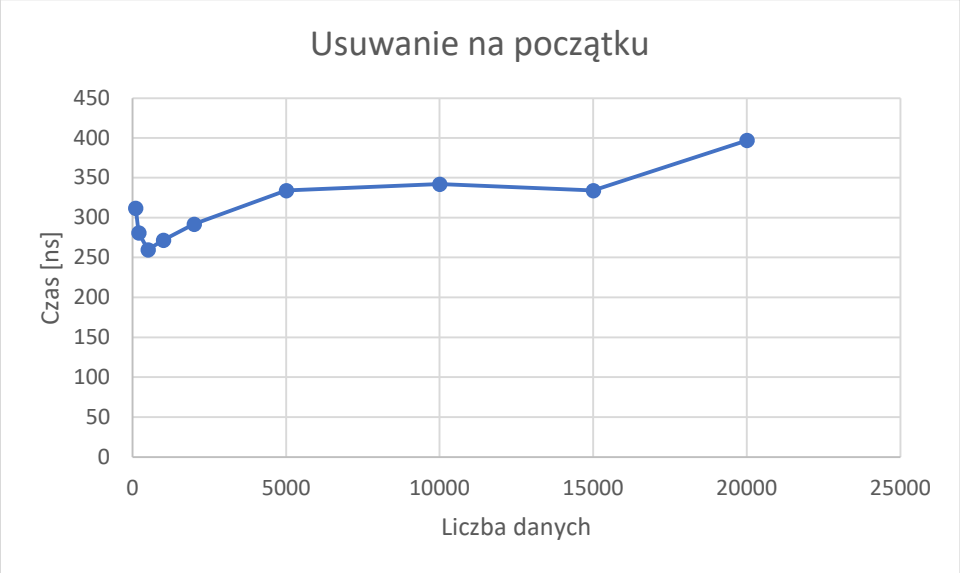
Wykres 23: Dodawanie elementu na koniec listy wypełnionej wartościami [1;max_int]



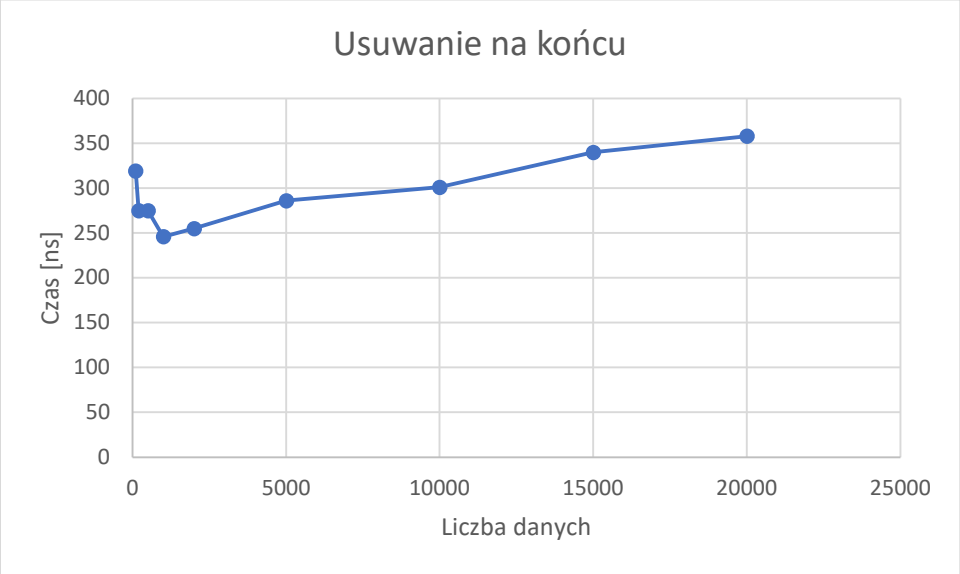
Wykres 24: Dodawanie elementu w środek listy wypełnionej wartościami [1;max_int]



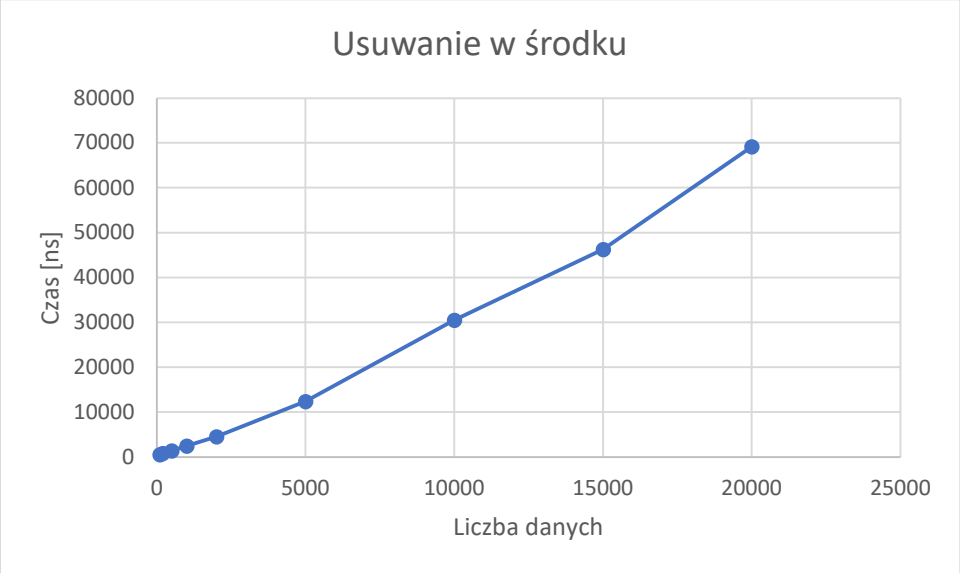
Wykres 25: Usuwanie elementu na początku listy wypełnionej wartościami [1;max_int]



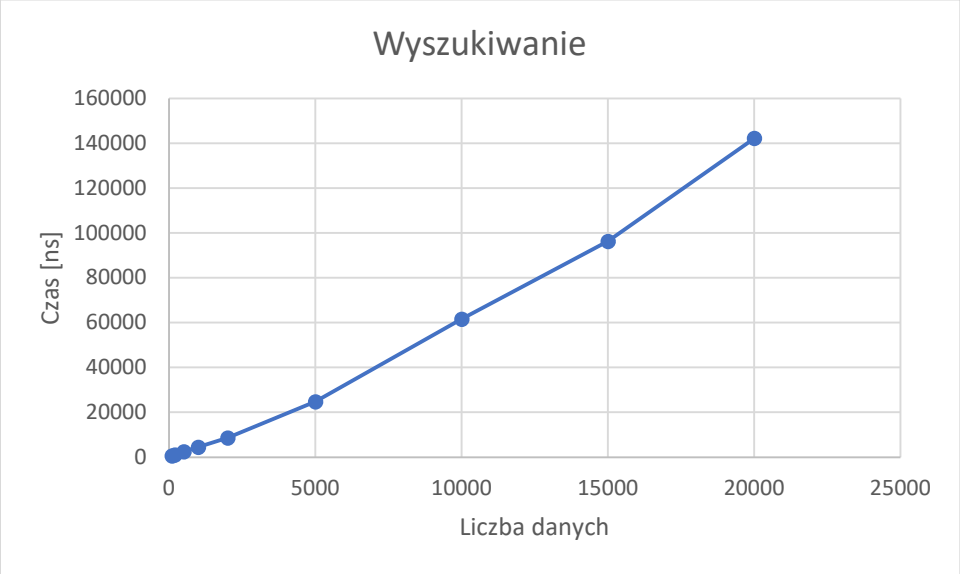
Wykres 26: Usuwanie elementu na końcu listy wypełnionej wartościami [1;max_int]



Wykres 27: Usuwanie elementu w środku listy wypełnionej wartościami [1;max_int]



Wykres 28: Wyszukiwanie elementu w liście wypełnionej wartościami [1;max_int]

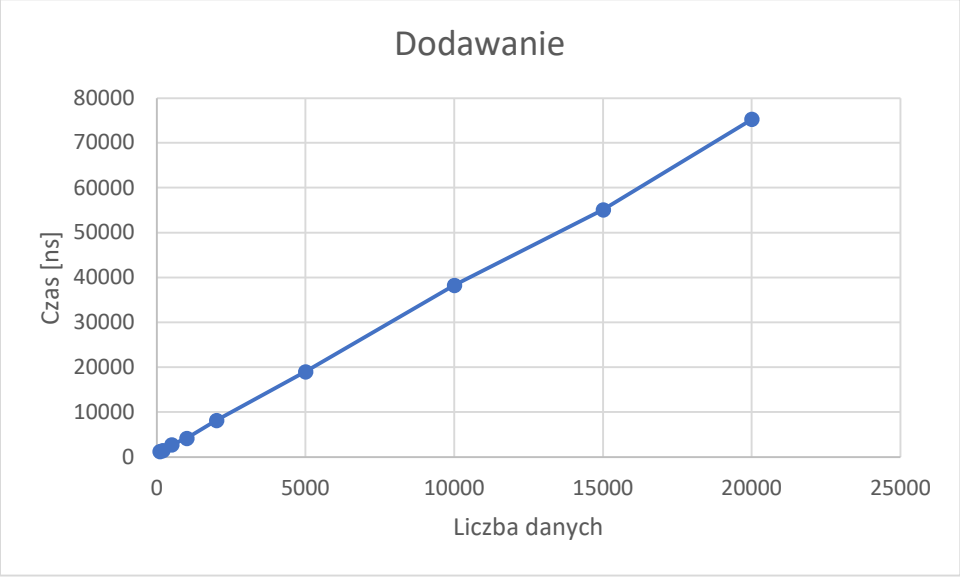


3.3 Kopiec binarny

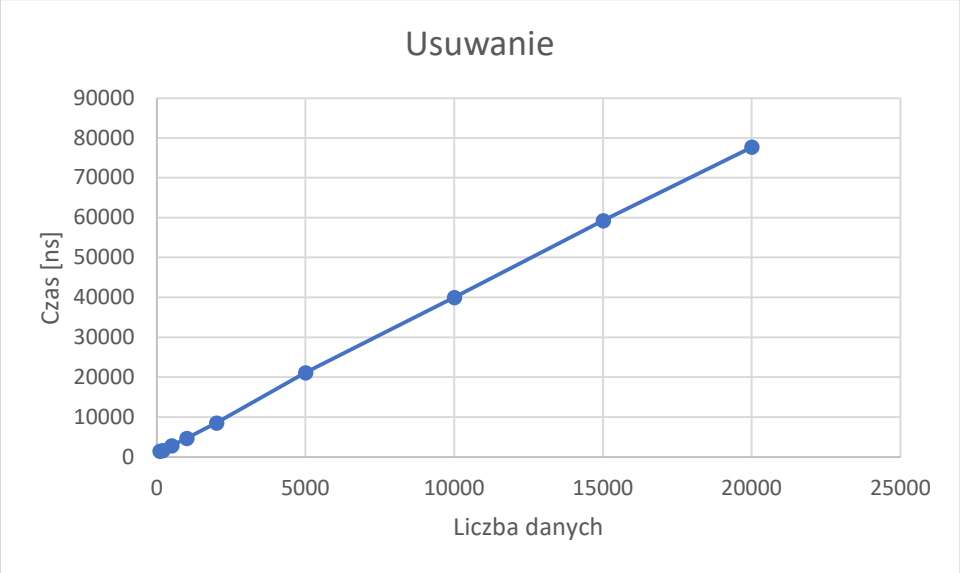
Tabela 5: Pomiary dla kopca binarnego wypełnionego wartościami z przedziału [1;100]

	Liczba elementów	Dodawanie	Usuwanie	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]
1	100	1231	1485	401
2	200	1459	1655	619
3	500	2687	2805	1368
4	1000	4123	4696	2630
5	2000	8155	8554	5097
6	5000	18991	21098	12956
7	10000	38285	40015	25302
8	15000	55084	59246	37584
9	20000	75300	77706	49834

Wykres 29: Dodawanie elementu do kopca wypełnionego wartościami [1;100]



Wykres 30: Usuwanie korzenia kopca wypełnionego wartościami [1;100]



Wykres 31: Wyszukiwanie elementu w kopcu wypełnionym wartościami [1;100]

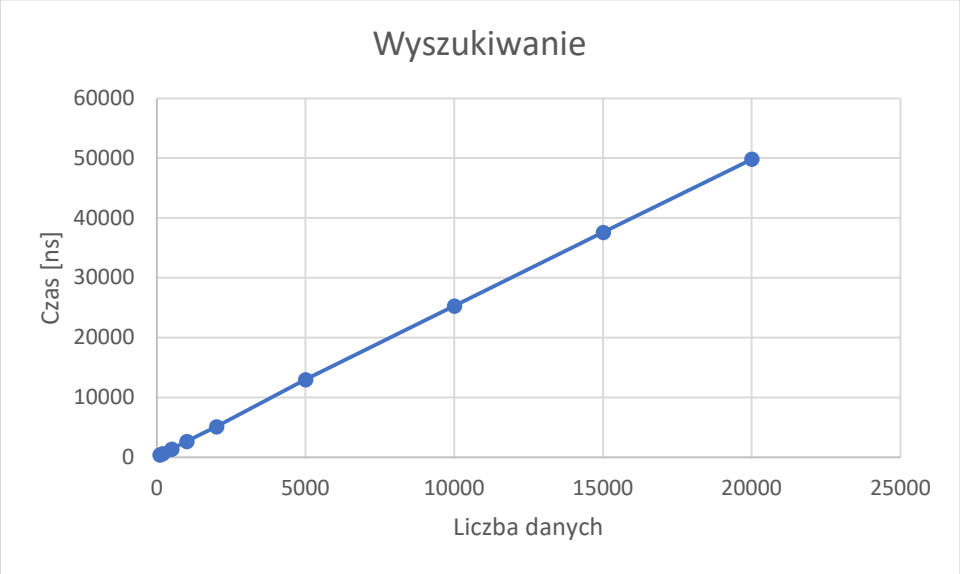
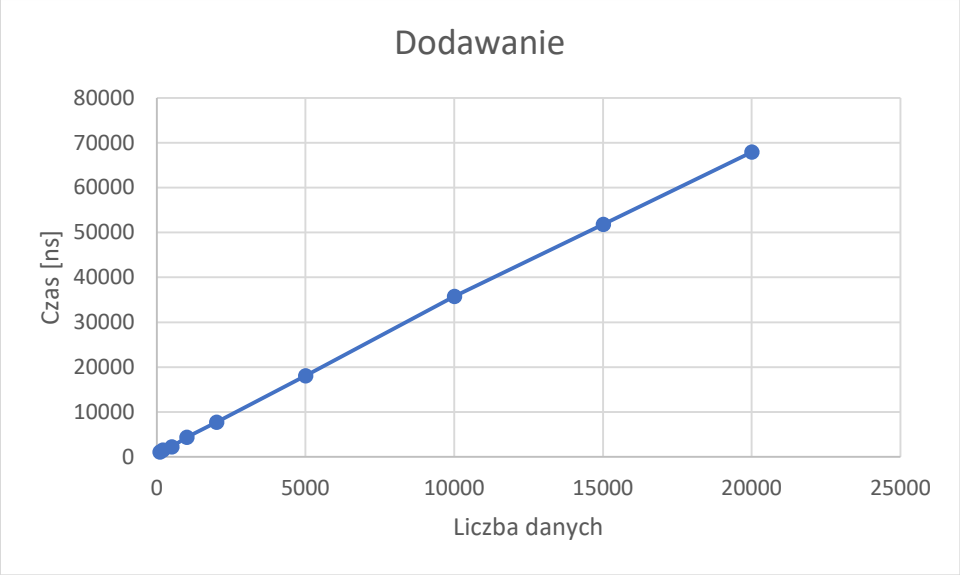


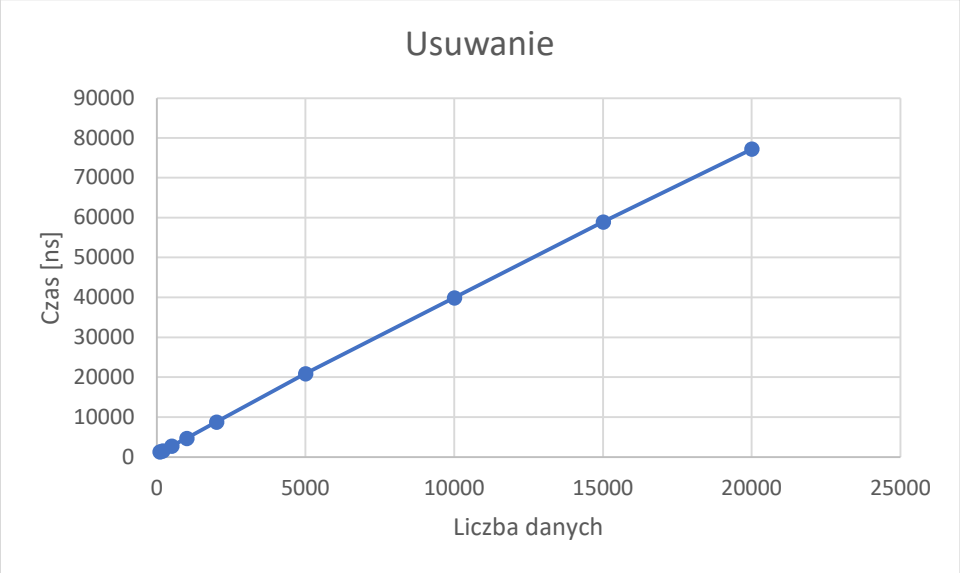
Tabela 6: Pomiary dla kopca binarnego wypełnionego wartościami z przedziału [1;max_int]

	Liczba elementów	Dodawanie	Usuwanie	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]
1	100	1056	1259	418
2	200	1464	1512	613
3	500	2237	2753	1373
4	1000	4352	4690	2635
5	2000	7725	8777	5187
6	5000	18045	20865	12744
7	10000	35741	39880	25158
8	15000	51775	58926	37410
9	20000	67956	77215	49532

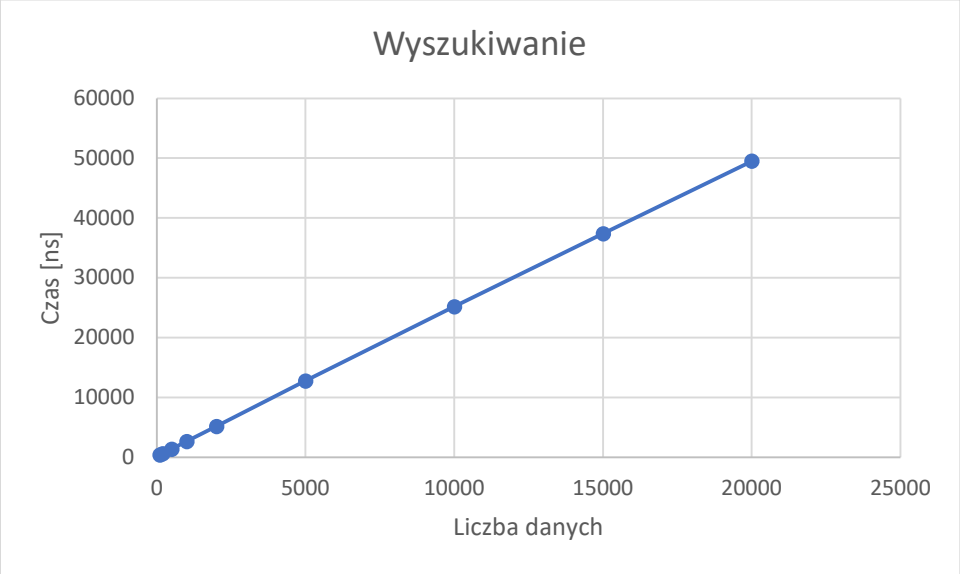
Wykres 32: Dodawanie elementu do kopca wypełnionego wartościami [1;max_int]



Wykres 33: Usuwanie korzenia kopca wypełnionego wartościami [1;max_int]



Wykres 34: Wyszukiwanie elementu w kopcu wypełnionym wartościami [1;max_int]

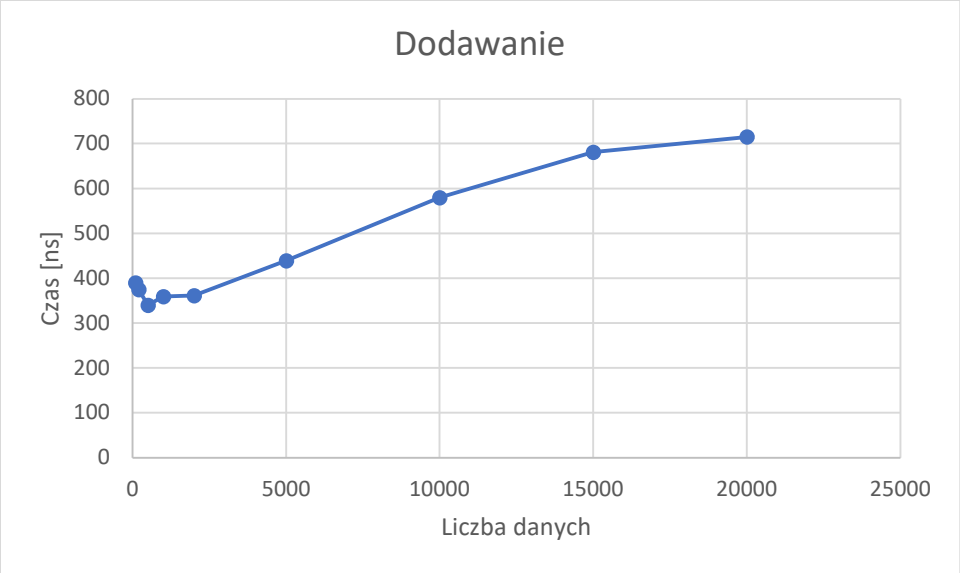


3.4 Drzewo czerwono-czarne

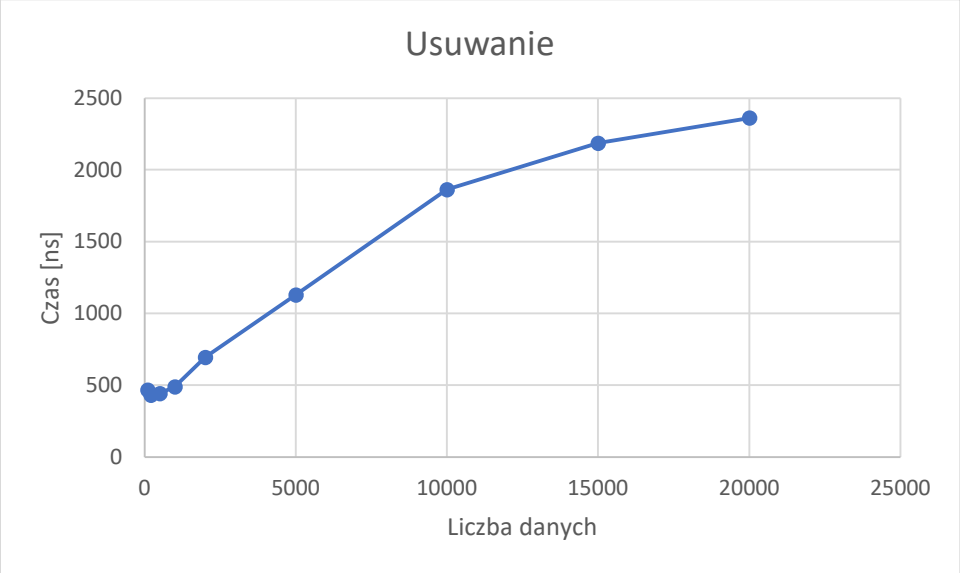
Tabela 7: Pomiary dla drzewa czerwono-czarnego wypełnionego wartościami z przedziału [1;20000]

	Liczba elementów	Dodawanie	Usuwanie	Wyszukiwanie
L.p.		[ns]	[ns]	[ns]
1	100	1231	1485	401
2	200	1459	1655	619
3	500	2687	2805	1368
4	1000	4123	4696	2630
5	2000	8155	8554	5097
6	5000	18991	21098	12956
7	10000	38285	40015	25302
8	15000	55084	59246	37584
9	20000	75300	77706	49834

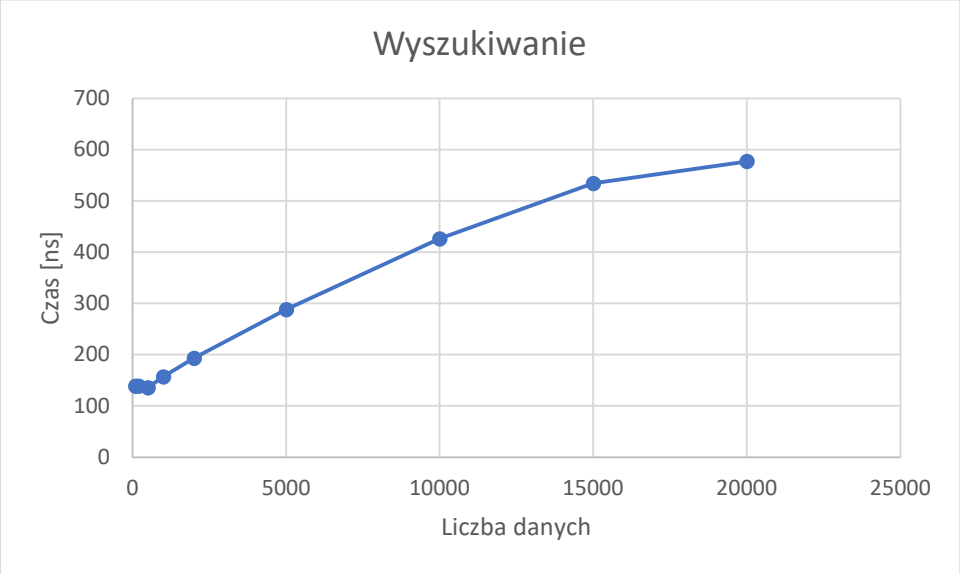
Wykres 35: Dodawanie elementu do drzewa wypełnionego wartościami [1;20000]



Wykres 36: Usuwanie węzła drzewa wypełnionego wartościami [1;20000]



Wykres 37: Wyszukiwanie elementu w drzewie wypełnionym wartościami [1;20000]



4 Wnioski

Dla tablicy dynamicznej czas wykonania poszczególnych operacji zależy od wielkości badanej struktury, na wykresach widzimy niemal idealnie proste linie co zgadza się z wartościami książkowymi, wartość elementów nie ma zauważalnego wpływu na wyniki. W liście dodawanie jak i usuwanie z początku i z końca struktury niezależnie od wartości ma stały czas wykonywania, natomiast wykresy pozostałych operacji przedstawiają zależność liniową, co również zgadza się z wartościami wzorcowymi.
Przy pomiarach dla kopca wykresy także mają postać liniową, co odbiega od oczekiwanych złożoności które mają postać logarytmiczną. Powodem tej rozbieżności jest sposób implementacji tej struktury. Złożoność książkowa odnosi się do kopca ze stałym rozmiarem struktury, natomiast w naszych założeniach każda struktura musiała być alokowana dynamicznie, przez co wystąpiła taka różnica.
Wykresy dla ostatniej struktury, czyli drzewa czerwono-czarnego mają postać logarytmiczną i zgadza się to ze złożonością książkową. Podsumowując, na podstawie otrzymanych danych, możemy wywnioskować, iż operacje na drzewie były znacząco szybsze niż na pozostałych strukturach, a przedział wartości kluczy nie ma widocznego wpływu na prędkość wykonywania poszczególnych operacji.