

Struktury Danych i Złożoność Obliczeniowa

Autor: Alicja Wróbel

1.Cel projektu:

Celem projektu było utworzenie struktur danych takich jak tablica, lista dwukierunkowa oraz kopiec typu max, na których można wykonać operacje: dodawania, usuwania oraz wyszukiwania elementów o zadanej wartości.

2.Złożność obliczeniowa:

Wyróżniamy 2 rodzaje złożoności pamięciowa oraz czasową. Złożoność pamięciowa określa nam ilość zużycia pamięci, która zostanie wykorzystana w celu danej operacji, natomiast czasowa to ilość czasu potrzebna na wykonanie danego algorytmu. Złożoności dzielimy na optymistyczne (możliwy najkrótszy czas wykonania operacji), średnie (średnia z uzyskanych pomiarów), pesymistyczna(możliwy najdłuższy czas wykonania operacji).

Operacje na tablicy:

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

Operacje na liście:

	Pesymistyczna
Dodanie wartości	$O(n)$
Usunięcie wartości	$O(n)$
Wyszukanie wartości	$O(n)$

Operacje na kopcu:

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

3.Plan eksperymentu:

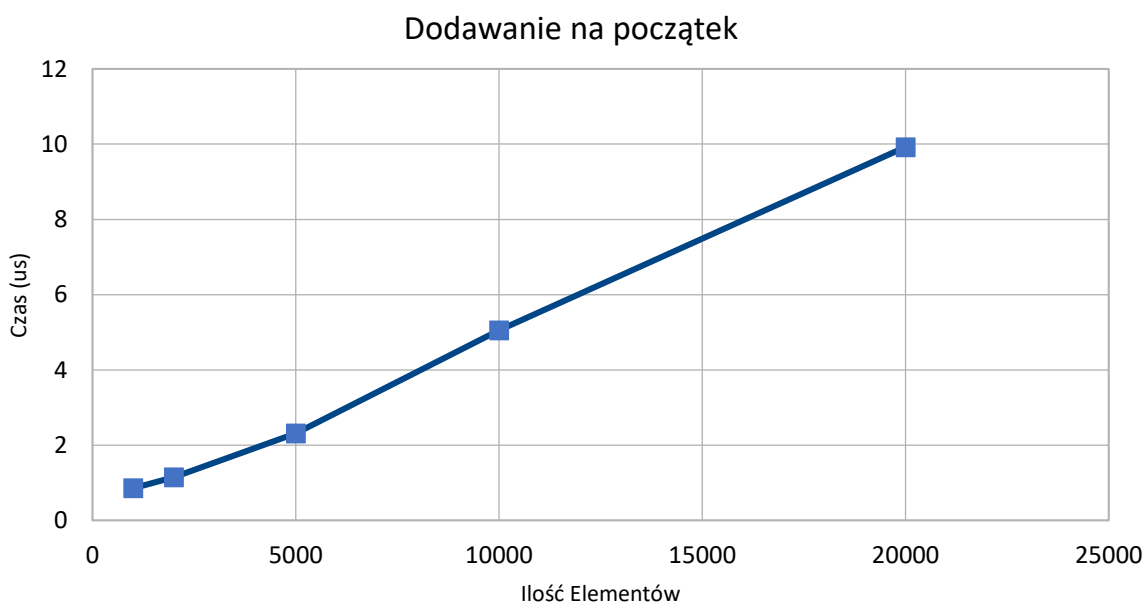
Do obliczeń została wykorzystana funkcja `QueryPerformanceCounter()`, w klasie „Timer.h”. Każdy pomiar został wykonany dla innego zestawu danych losowych dla struktury, gdzie wartości wahają się od -1000 do 1000. W przedstawionych poniżej pomiarach jest wartość średnia 100 pomiarów dla różnej ilości parametrów: 1000, 2000, 5000, 10000 i 20000.

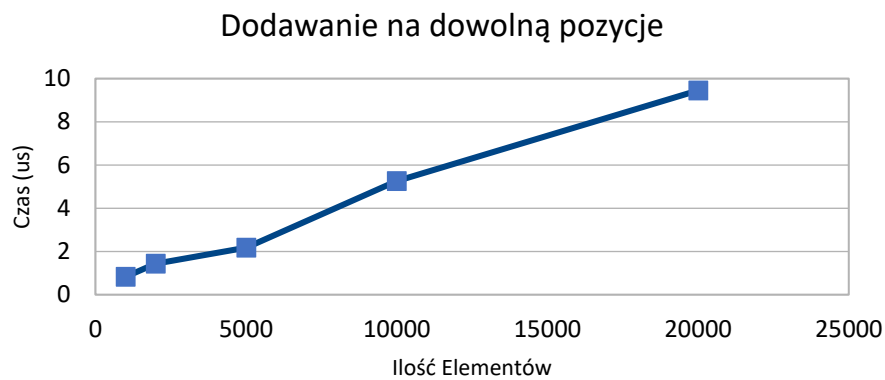
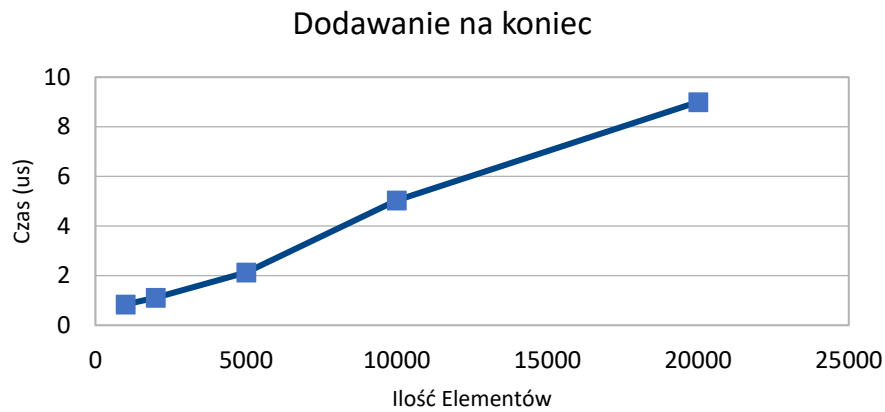
4.Struktury danych:

4.1.Tablica

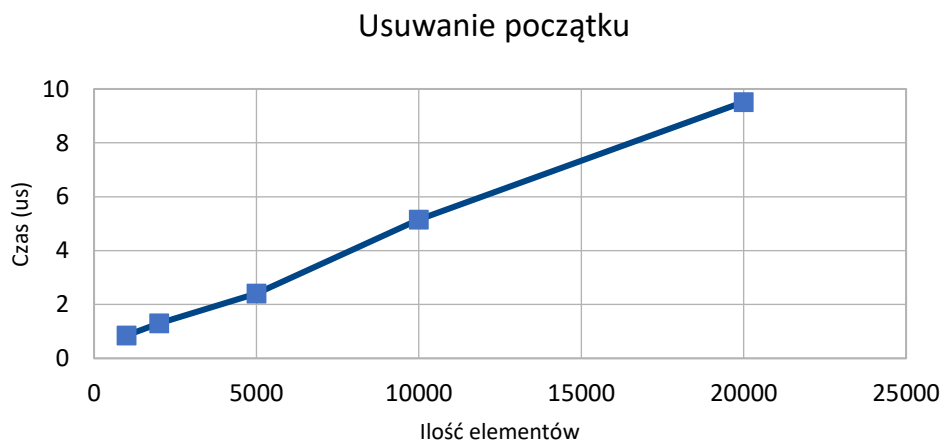
Tablica jest to struktura danych, której elementy zawierają wartość oraz indeks, w zależności od położenia w strukturze. Przedstawiona tablica jest dynamiczna ponieważ wykorzystuje alokację pamięci. Rezerwuje miejsce w pamięci przed dodaniem oraz zwalnia je po usunięciu elementu.

Do usunięcia elementu wymagana jest alokacja pamięci o 1 większy niż rozmiar tablicy. Wartość jej jest przypisywana do tablicy pomocniczej oraz dodanie w danym miejscu nowego elementu, a pamięć tablicy głównej jest zwalniana. Następnie wartości tablicy pomocniczej są przypisane do tablicy głównej. Rozmiar tablicy jest zwiększony o 1.

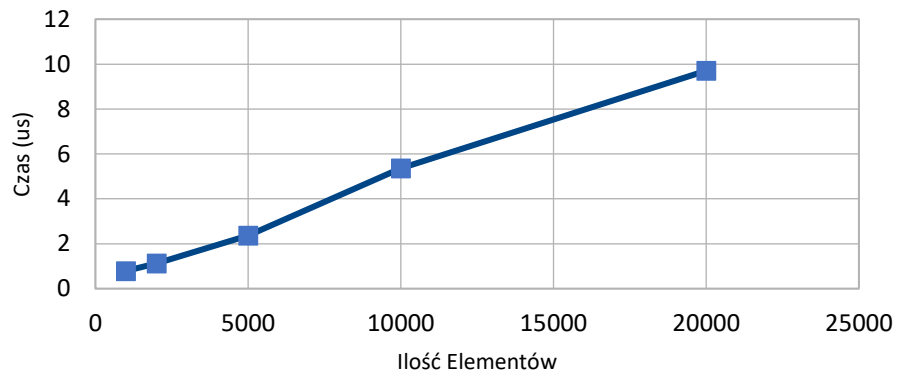




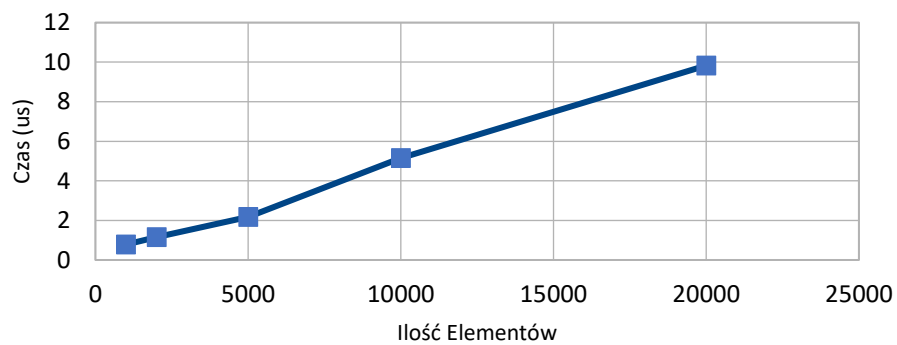
Do usunięcia elementu wymagana jest alokacja pamięci o 1 mniejszy niż rozmiar tablicy. Wartość jej jest przypisywana do tablicy pomocniczej bez wybranego elementu, a pamięć tablicy głównej jest zwalniana. Następnie wartości tablicy pomocniczej są przypisane do tablicy głównej. Rozmiar tablicy jest zmniejszony o 1.



Usuwanie końca

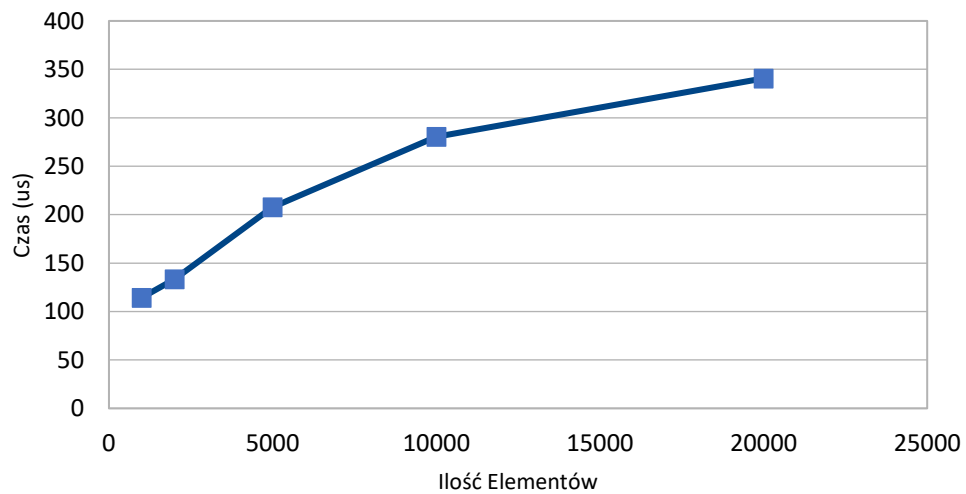


Usuwanie z dowolnej pozycji



Wyszukiwanie polega na iteracji od pierwszego elementu, gdy znajduje zdaną wartość wyświetla jej indeks.

Wyszukiwanie

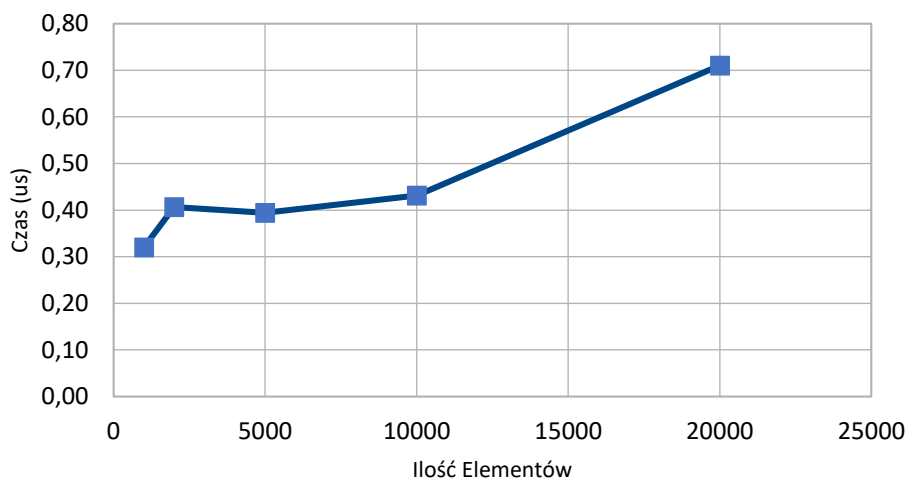


4.2. Lista dwukierunkowa

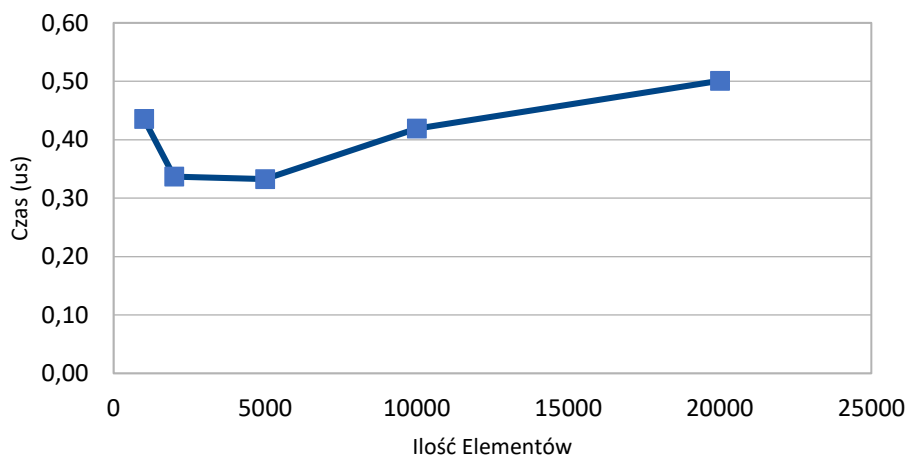
Lista jest to struktura danych, której każdy element ma 2 wskaźniki, na następnika oraz poprzednika. Dzięki temu dostęp do poszczególnych elementów jest sekwencyjny. Pierwszy element czyli head jest to element, który nie posiada poprzednika, a więc wskaźnik wynosi nullptr. Tak samo jest w przypadku ostatniego elementu czyli tail, który natomiast nie posiada następnika. W przeciwieństwie do tablicy, lista nie wymaga ciągłego obszaru pamięci.

Dodawanie jak i usuwanie na liście polega na zmienianiu odpowiednio wskaźnika poprzednika i następnika elementów znajdujących się pomiędzy dodawanym lub usuwanym elementem.

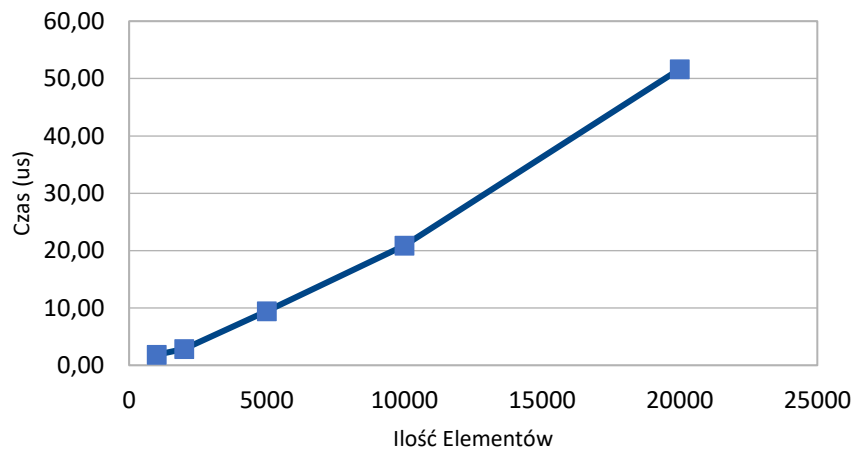
Dodawanie na początek



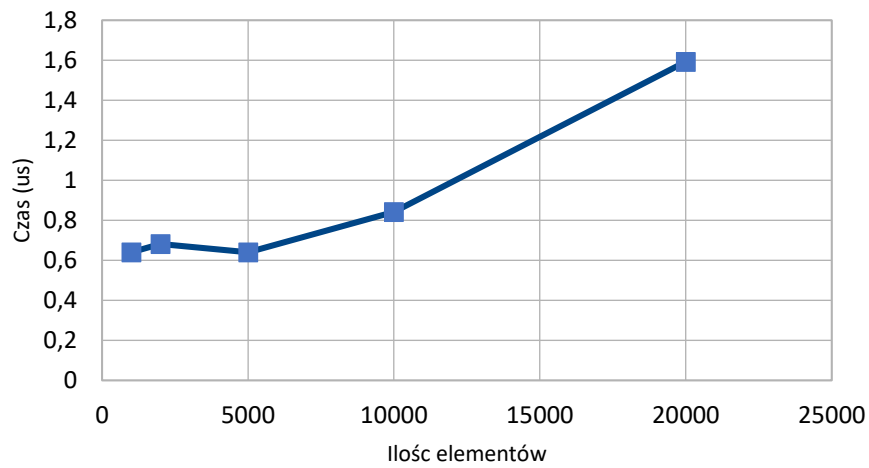
Dodawanie na koniec



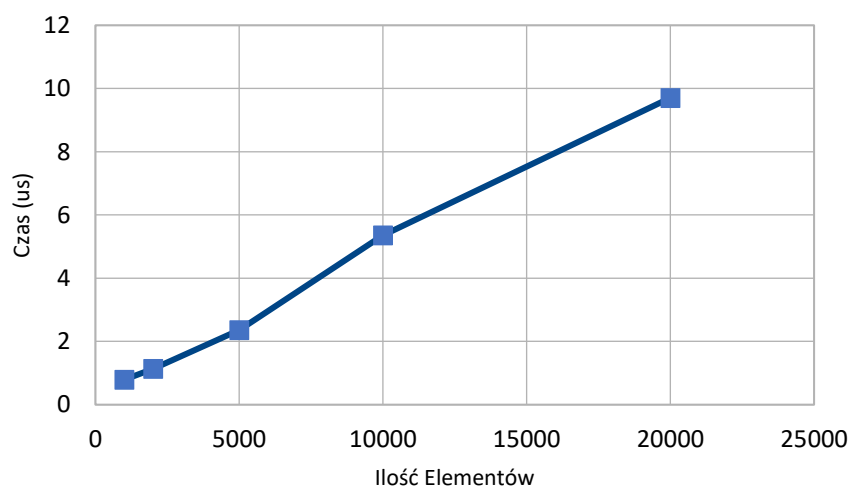
Dodawanie na dowolna pozycje

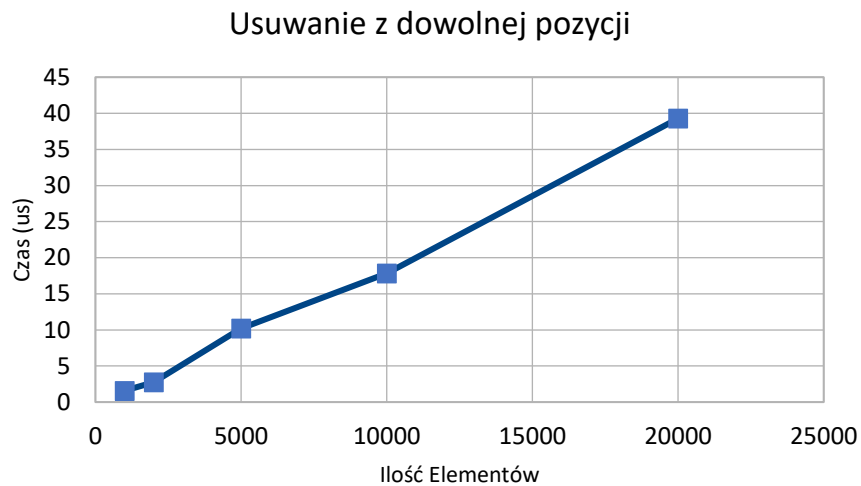


Usuwanie początku

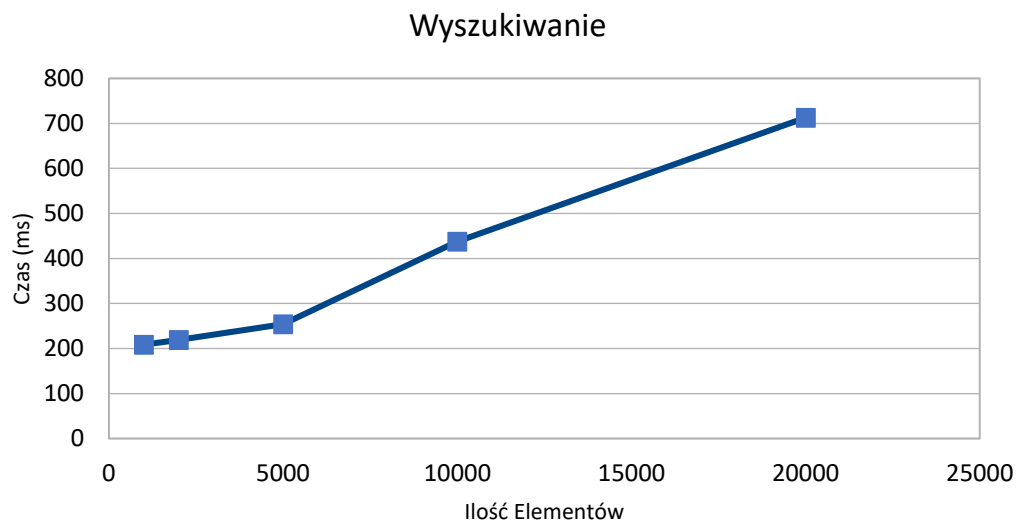


Usuwanie z końca





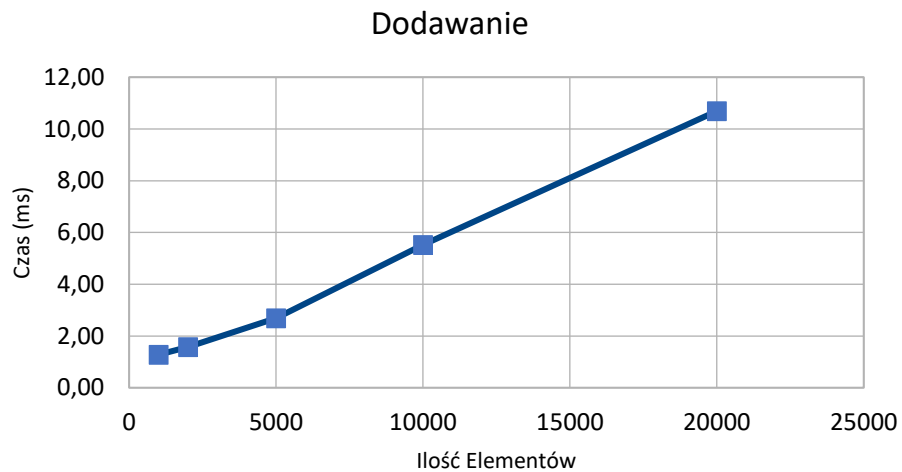
Wyszukiwanie natomiast polega na iteracji od elementu head do tail.



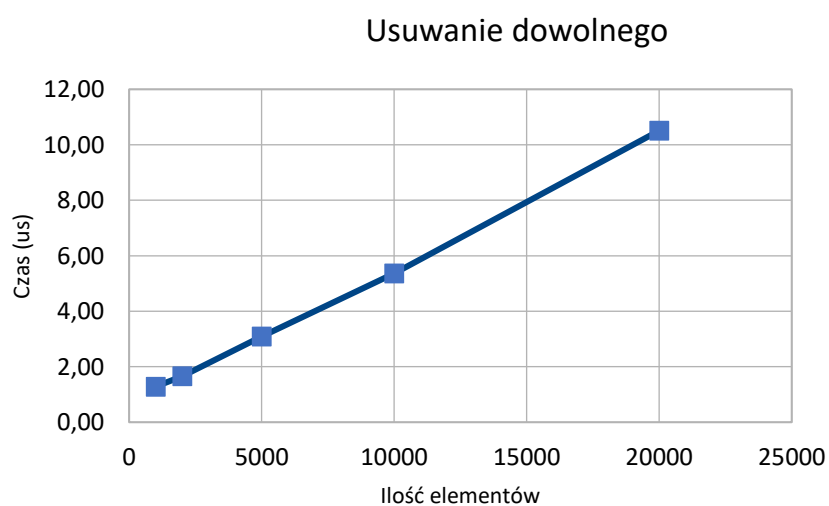
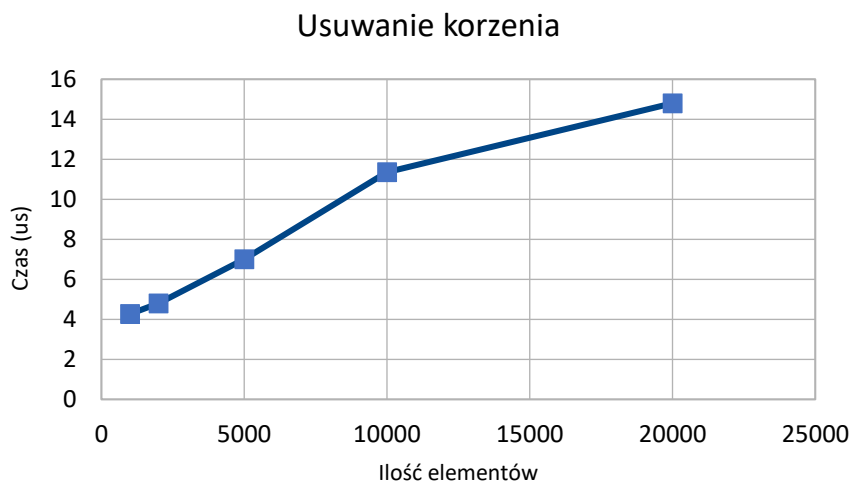
4.3. Kopiec typu max

Kopiec jest kompletnym drzewem binarnym co oznacza że może posiadać 0, 1 lub 2 synów oraz patrząc od strony lewej do prawej nie posiada przerw między węzłami. Żaden z synów nie przechowuje wartości większej od ojca. Przez co każde dodanie elementu lub jego usunięcie wymaga sortowania.

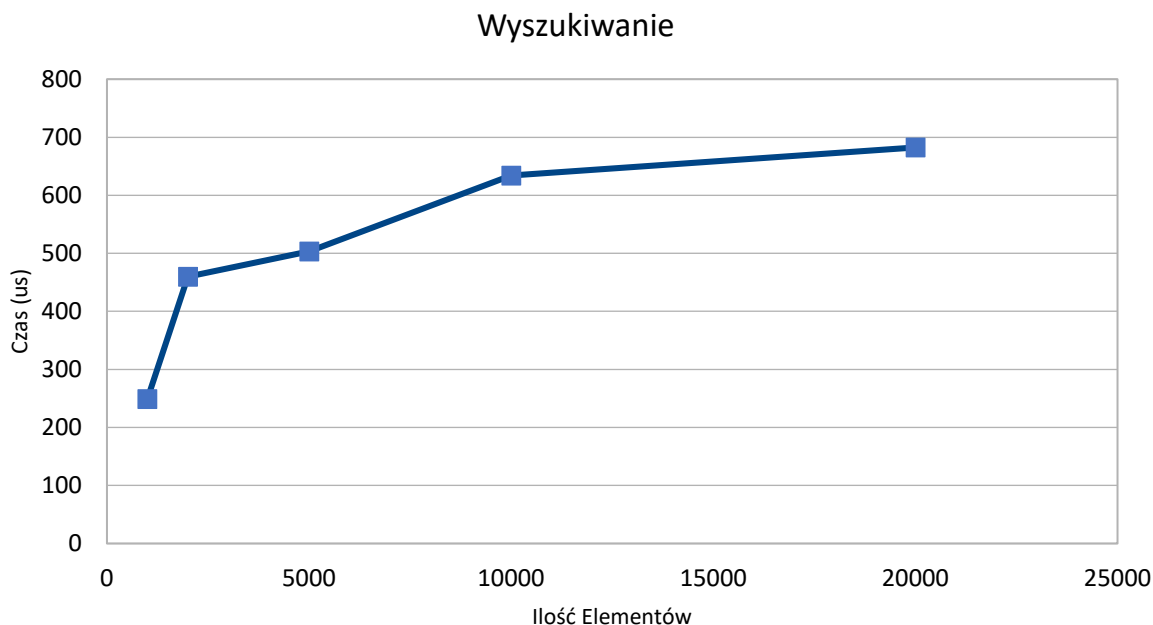
Dodawanie w kopcu polega na wstawieniu nowego elementu na koniec, w razie nie zachowania własności kopca wykonuje się jego przywracanie.



Usunięcie elementu polega na usunięciu danej wartości następnie w jej miejscu wpisanie ostatniego liścia kopca i przywróceniu własności.



Wyszukiwanie w kopcu jest funkcją rekurencyjną, wartość klucza jest porównywana z węzłem i jego potomkiem.



5. Wnioski:

Operacje na tablicy potwierdzają złożoność $O(n)$. Wynika to z faktu, że za każdym razem potrzebna jest relokacja pamięci.

Przy liście widzimy już większe odstępstwo od teorii co do danej struktury. Przy dodawaniu elementów na początek, koniec i przy usunięciu początku różnica w czasie jest zbliżona do siebie $O(1)$. Wyszukiwanie, dodanie elementu lub jego usunięcie w dowolnym miejscu wynosi $O(n)$ zgodnie z prawdą. Natomiast różnice występują przy usunięciu elementu z końca, której złożoność wynosi $O(n)$.

Odstępstwo od prawdy widać jednak niestety również w kopcu. Wyniki w większości wskazują bardziej na złożoność obliczeniową równą $O(n)$ niż $O(\log n)$. Błąd może wynikać z tego, że dany kopiec został utworzony na własno utworzonej strukturze tablicowej.

Mimo błędów program realizuje poprawne działanie struktur.

W przypadku dodawania, usuwania elementu na koniec jak i początek lista jest bardziej wydajna niż tablica.

Tablica							
Czas (us)							
Ilość elementów	Dodawanie			Usuwanie			Wyszukiwanie
	Na początek	Na koniec	Dowolnie	Początku	Końca	Dowolnie	
1000	0,853936	0,84162	0,825198	0,849831	0,788249	0,780038	114,276
2000	1,14953	1,11258	1,43691	1,29733	1,129	1,15774	133,44
5000	2,31548	2,13484	2,18	2,40991	2,36064	2,18	207,576
10000	5,04972	5,04151	5,26064	5,15646	5,36064	5,15236	280,498
20000	9,92	9,00328	9,45078	9,51646	9,70531	9,82848	340,668

Lista							
Czas (us)							
Ilość elementów	Dodawanie			Usuwanie			Wyszukiwanie
	Na początek	Na koniec	Dowolnie	Początku	Końca	Dowolnie	
1000	0,32	0,44	1,84	0,640452	0,410546	1,5806	208,46
2000	0,41	0,34	2,85	0,681507	0,53371	2,73834	218,907
5000	0,39	0,33	9,45	0,640452	0,944256	10,21	253,796
10000	0,43	0,42	20,90	0,84162	0,956573	17,8547	437,712
20000	0,71	0,50	51,65	1,59292	1,38765	39,277	712,721

Kopiec				
Czas (us)				
Ilość elementów	Dodawanie	Usuwanie		Wyszukiwanie
		Korzenia	Dowolnie	
1000	1,28	4,26968	1,28	208,46
2000	1,56829	4,79518	1,65861	218,907
5000	2,68908	7,00392	3,09	253,796
10000	5,52185	11,35	5,36584	437,712
20000	10,69	14,8002	10,51	712,721

