Jakub Plona

Struktury Danych i Złożoność Obliczeniowa

Zadanie projektowe nr 1: Badanie efektywności operacji dodawania, usuwania oraz wyszukiwania elementów w różnych strukturach danych

07/04/2019

Spis treści

1. Wstęp teoretyczny

Przeprowadzony eksperyment miał na celu zbadanie efektywności podstawowych operacji (często nazywanych też słownikowymi) na strukturach danych takich, jak tablica dynamiczna, lista (dwukierunkowa z wartownikiem), kopiec (maksymalny) oraz drzewo czerwono-czarne. Są to: dodawanie elementu do struktury, usuwanie elementu ze struktury oraz wyszukiwanie elementu w strukturze.

Do opisu czasu działania tychże operacji posłużono się asymptotyczną notacją dużego O.

1.1 Tablica dynamiczna

Tablica jest reprezentowana w pamięci komputera jako spójny obszar pamięci. Aby uzyskać dostęp do takiej struktury, koniecznym jest posiadanie adresu jej początku. Z założenia elementy w tablicy są tego samego typu, dlatego dostęp do elementu jest natychmiastowy (gdy znamy jego indeks), gdyż adres tego elementu jest wyliczany na podstawie wzoru:

adres początkowy + rozmiar elementu * indeks elementu

W przeprowadzonych badaniach zaimplementowano i posłużono się tablicą dynamiczną o zamortyzowanym koszcie operacji na jej końcu równym O(1).

- Wstawianie

  Zaimplementowana tablica dynamiczna zwiększa swój rozmiar dwukrotnie, jeżeli podczas próby wstawienia elementu okaże się, że struktura jest pełna. Zamortyzowany koszt tej operacji wynosi O(1).

  - Początek tablicy

    Jeżeli tablica nie jest pusta, to wszystkie jej elementy są przesuwane o jedną pozycję do przodu. Po operacji warunkowej element jest wstawiany na pozycję o indeksie 0. Złożoność tej operacji z racji przesuwania tablicy wynosi O(n).

  - Koniec tablicy

    Element zostaje dodany na koniec tablicy po jej ewentualnym rozszerzeniu. Koszt tej operacji wynosi O(1).

  - Losowe miejsce tablicy

    Element zostaje wstawiony na losowo wybraną pozycję tablicy (elementy od wybranej pozycji począwszy są przesuwane w prawo o jedno miejsce). Koszt tej operacji wynosi O(n).

- Usunięcie

  Zaimplementowana tablica dynamiczna zmniejsza swój rozmiar dwukrotnie, jeżeli podczas próby usunięcia elementu okaże się, że struktura jest w ¾ pusta. Zamortyzowany koszt tej operacji wynosi O(1).

  - Początek tablicy

    Jeżeli tablica nie jest pusta, to wszystkie jej elementy są przesuwane o jedną pozycję do tyłu. Złożoność tej operacji z racji przesuwania tablicy wynosi O(n).

- Koniec tablicy

  W tym przypadku zostaje zmniejszony jedynie rozmiar tablicy. Koszt tej operacji wynosi O(1).

- Losowe miejsce tablicy

  Element zostaje usunięty z losowo wybranej pozycji tablicy poprzez przesunięcie elementów dalszych o jedną pozycję w lewo. Koszt tej operacji wynosi O(n).

- Wyszukiwanie

  Zaimplementowano klasyczne wyszukiwanie liniowe. Koszt znalezienia elementu jest opisywany przez O(n).

1.2. Lista dwukierunkowa

List dwukierunkowa ze strażnikiem jest wskaźnikową strukturą danych. Kolejno dodawane do niej elementy poza kluczem zawierają jeszcze wskaźniki na elementy: poprzedni, następny. Wartownik jest specjalnym elementem tablicy zawsze w niej obecnym – jest to element pośredni pomiędzy pierwszym, a ostatnim wpisem na liście. Z racji obecności wartownika, lista jest tak naprawdę cykliczna, gdyż strażnik łączy ze sobą element pierwszy i ostatni.

Koszt obsługi strażnika jest określony przez O(1).

- Wstawianie

  - Początek listy

    Operacja ta polega na przepisaniu wskaźników w taki sposób, aby wstawiany element był wskazywany jako następnym przez strażnika, zaś poprzedni pierwszy element był zaraz za nowo wstawianym. Złożoność tej operacji wynosi O(1).

  - Koniec listy

    Operacja ta jest identyczna (z dokładnością do symetrii) z wstawianiem na początek listy, gdyż ostatni element jest wskazaniem poprzedniego w polu strażnika. Złożoność tej operacji wynosi O(1).

  - Losowe miejsce na liście

    Różnica między wstawianiem w losowe miejsce a wstawianiem na początek jest taka, że w tym wypadku element bazowy najpierw trzeba znaleźć (nie jest już nim strażnik). Złożoność tej operacji wynosi O(n)

- Usunięcie

  - Początek listy

    Operacja ta polega na przepisaniu wskaźników w taki sposób, aby następnik i poprzednik (strażnik) usuwanego elementu był ze sobą połączone. Złożoność tej operacji wynosi O(1).

- Koniec listy

  Operacja ta jest identyczna (z dokładnością do symetrii) z usuwaniem z początku listy. Różnica jest taka, iż strażnik jest teraz następnikiem. Złożoność tej operacji wynosi O(1).

- Losowe miejsce na liście

  Przed wykonaniem tej operacji należy najpierw znaleźć element do usunięcia. Pozostała część algorytmu jest taka sama, jak w przypadku usuwania z końca listy. Złożoność tej operacji wynosi O(n).

- Wyszukanie

  Aby znaleźć element należy przeszukać całą listę. Złożoność tej operacji wynosi O(n).

1.3. Kopiec maksymalny

Kopiec jest tablicową strukturą danych która posiada tzw. własność kopca, zaś jej elementy są indeksowane w specyficzny sposób. Dla kopca maksymalnego własność kopca brzmi następująco: wartość przodka jest niemniejsza od wartości potomka. Indeksowanie elementów polega na wyliczaniu indeksów rodziców i dzieci węzła na podstawie określonych wzorów. I tak dla rodzica jest to |_i/2_|, dla lewego dziecka 2i, zaś dla prawego dziecka 2i + 1, gdzie i jest indeksem węzła z zakresu [1…n]. Dzięki takiemu indeksowaniu kopiec można przedstawiać w formie drzewa. Do implementacji kopca użyto zaimplementowanej wcześniej tablicy dynamicznej.

- Wstawianie

  Wstawienie elementu do kopca polega na dołączeniu go na koniec i pięciu się w górę drzewa w celu znalezienia takiej pozycji dla nowego węzła, że wstawiając go na nią zachowana będzie własność kopca.
  Złożoność tej operacji wynosi O(h), gdzie h oznacza wysokość kopca liczoną ze wzoru: lgn (gdzie n – liczba elementów w kopcu).

- Usuwanie

  Operacja ta polega na przepisaniu ostatniego węzła do usuwanego i przywróceniu własności kopca (element może poruszać się w górę, bądź w dół zależnie od sytuacji).
  Złożoność tej operacji wynosi O(h).

- Wyszukiwanie

  Kopiec musi być przeszukany w całości, przy usprawnieniu, iż nie ma potrzeby przeszukiwać poddrzew węzła jeżeli jego wartość jest mniejsza od poszukiwanego elementu (w kopcu maksymalnym głębiej znajdziemy jedynie elementy mniejsze z uwagi

na własność kopca). Złożoność tej operacji wynosi jednak O(n), gdyż usprawnienie może jedynie zmienić stały współczynnik w funkcji złożoności obliczeniowej.

1.4. Drzewo czerwono-czarne

Drzewo czerwono-czarne jest drzewem poszukiwań binarnych, które spełnia tzw. warunki drzewa czerwono-czarnego.
Warunki te są następujące:

1) Wszystkie węzły w drzewie są czerwone albo czarne
2) Korzeń drzewa jest zawsze czarny
3) Liście drzewa są zawsze czarne
4) Jeśli węzeł jest czerwony, to obaj jego synowie są czarni
5) Każda prosta ścieżka od danego węzła do jego liści potomnych zawiera tę samą liczbę węzłów czarnych.

Drzewo poszukiwań binarnych spełniających ww. warunki jest drzewem zrównoważonym. W przypadku drzewa czerwono-czarnego oznacza to, że jego wysokość nie przekroczy dwukrotnej wartości wysokości minimalnej.

- Wstawianie

  Wstawienie elementu do drzewa czerwono-czarnego polega na wstawieniu elementu zgodnie z algorytmem wstawiania dla drzewa poszukiwań binarnych oraz przywrócenia warunków drzewa czerwono-czarnego. Czas potrzebny na wykonanie wstawienia do drzewa poszukiwań binarnych jest ograniczony przez O(lgn) (n – liczba elementów w drzewie), zaś przywrócenie warunków drzewa jest wykonywane w czasie stałym. Dlatego złożoność obliczeniowa tej operacji wynosi O(lgn).

- Usuwanie

  Operacja ta polega na usunięciu elementu z drzewa za pomocą algorytmu usuwania elementu z drzewa poszukiwań binarnych oraz przywróceniu własności drzewa czerwono-czarnego. Tak jak w przypadku wstawiania pierwsza część operacji jest wykonywana w czasie O(lgn), zaś druga jest ograniczona przez O(1).
  Złożoność tej operacji wynosi więc O(lgn).

- Wyszukiwanie

  Wyszukiwanie w drzewie czerwono-czarnym jest tym samym algorytmem, który jest stosowany w przypadku drzewa poszukiwań binarnych. Polega na przemieszczaniu się w głąb drzewa na podstawie decyzji wynikłych z porównań poszukiwanego klucza i klucza w aktualnym węźle (jest to możliwe ze względu na warunek drzewa poszukiwań binarnych: lewy potomek jest mniejszy niż przodek, zaś prawy potomek jest niemniejszy niż przodek).
  Złożoność tej operacji wynosi O(lgn).

2. Plan eksperymentu

- Wykorzystany język programowania to C++11

- Badanie poszczególnych operacji zostało przeprowadzone na losowo generowanych zestawach danych zawierających klucze z przedziałów: *[0, INT_MAX/2]*, *[INT_MAX/2, INT_MAX]*, *[0, INT_MAX]*, *[0,100] oraz [INT_MAX – 100, INT_MAX]*

- Losowe liczby z podanych przedziałów były generowane przy użyciu biblioteki *random*

- Badania przeprowadzano dla kolejnych rozmiarów struktur: *0, 1, 2, …, 20000*

- Ilość powtórzeń wykonania każdej operacji dla danego rozmiaru i wyniosła 100 (generowano nową populację)

- Otrzymanym wynikiem była suma zmierzonych czasów ze wszystkich powtórzeń dla danej operacji podzielona przez ilość powtórzeń (średni czas dla wykonanej operacji przy ustalonym rozmiarze dla różnych populacji)

- Otrzymane wyniki zaprezentowano w postaci surowej (pełne dane na wykresie) oraz uwzględniając jedynie tendencje wzrostowe danych (wykorzystano linie trendu)

- Czas wykonania operacji mierzony był przy użyciu funkcji QueryPerformanceFrequency() oraz QeryPerformanceCounter()

- Wszystkie struktury były alokowane dynamicznie

3. Wyniki

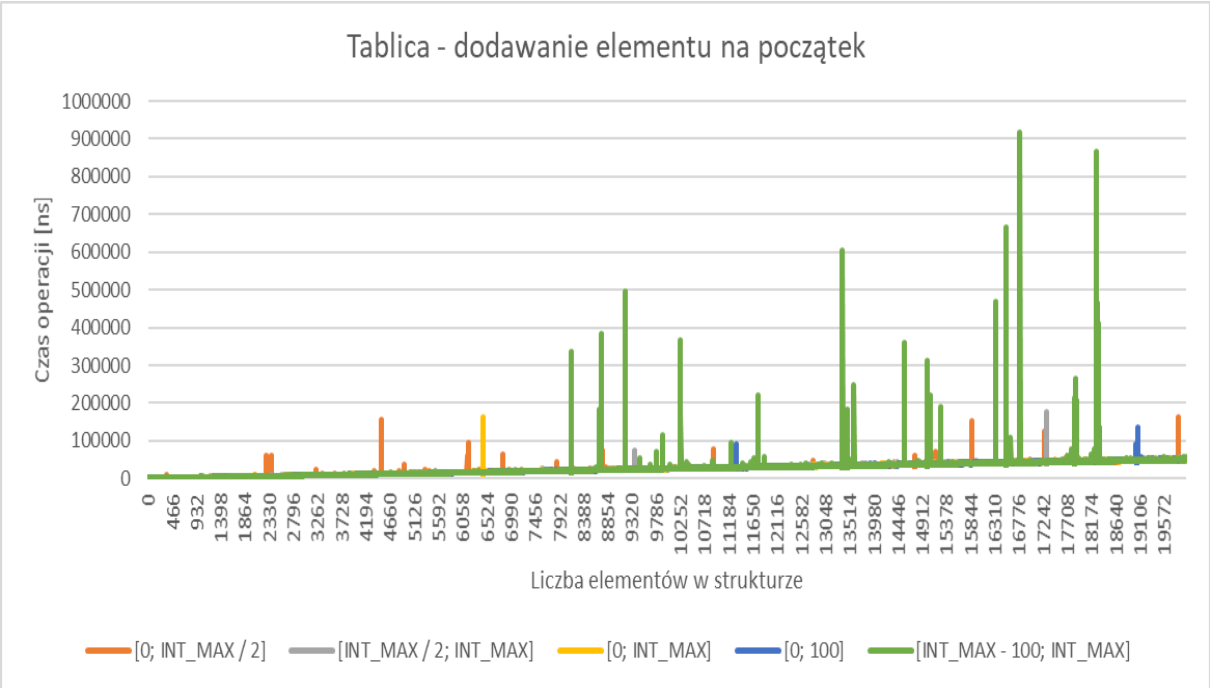
Czasy wykonywania poszczególnych operacji podane zostały w nanosekundach.

3.1. Tablica

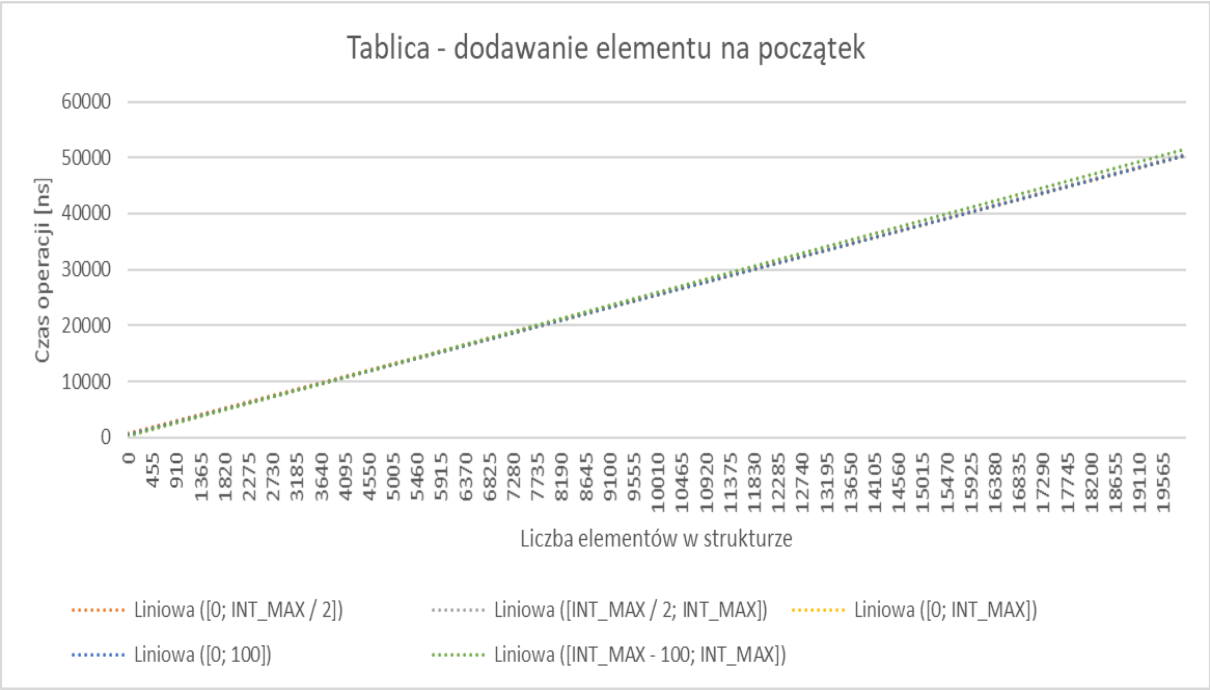
3.1.1. Wstawianie elementu na początek

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	1138	1095	1108	977	940
1000	3117	2870	2889	2854	3067
2000	5771	5323	5333	5299	5301
3000	8077	8067	8120	8207	8411
4000	10078	10419	11054	10299	10260
5000	13076	13280	12564	13456	12559
6000	15148	15330	14913	15253	15707
7000	17697	18273	17812	17606	18482
8000	20942	20699	20016	20441	20629
9000	24888	22918	24196	22787	23056
10000	25268	24771	25459	25976	25830
11000	29868	28154	28791	28087	27858
12000	30625	29655	29048	31306	31139
13000	33153	32565	33742	32147	32695
14000	34567	35247	34240	35037	36038
15000	38972	39358	37698	39253	37135
16000	40705	40328	40558	41322	39205
17000	41514	41414	41633	41046	41897
18000	45363	46039	44894	44547	45178
19999	50683	49340	50569	50222	49543

Tabela 1 Czas [ns] wstawiania elementu na początek tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 1 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu na początek tablicy



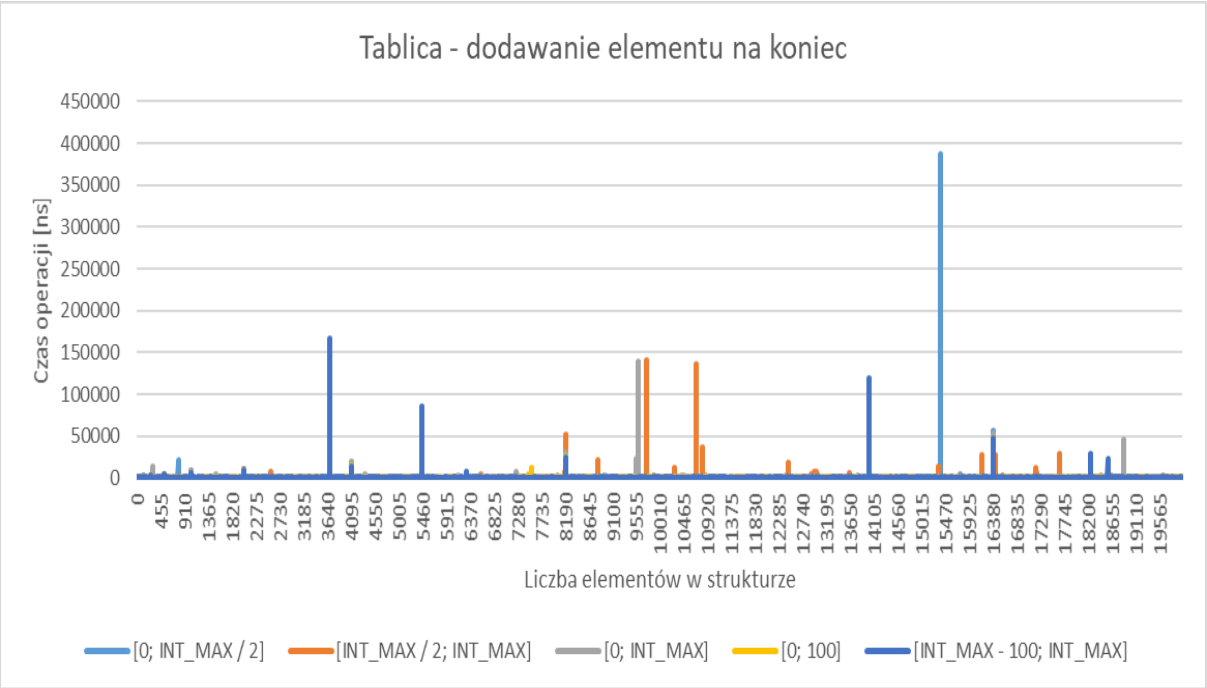
Rysunek 2 Linie trendu dla operacji wstawiania elementu na początek tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

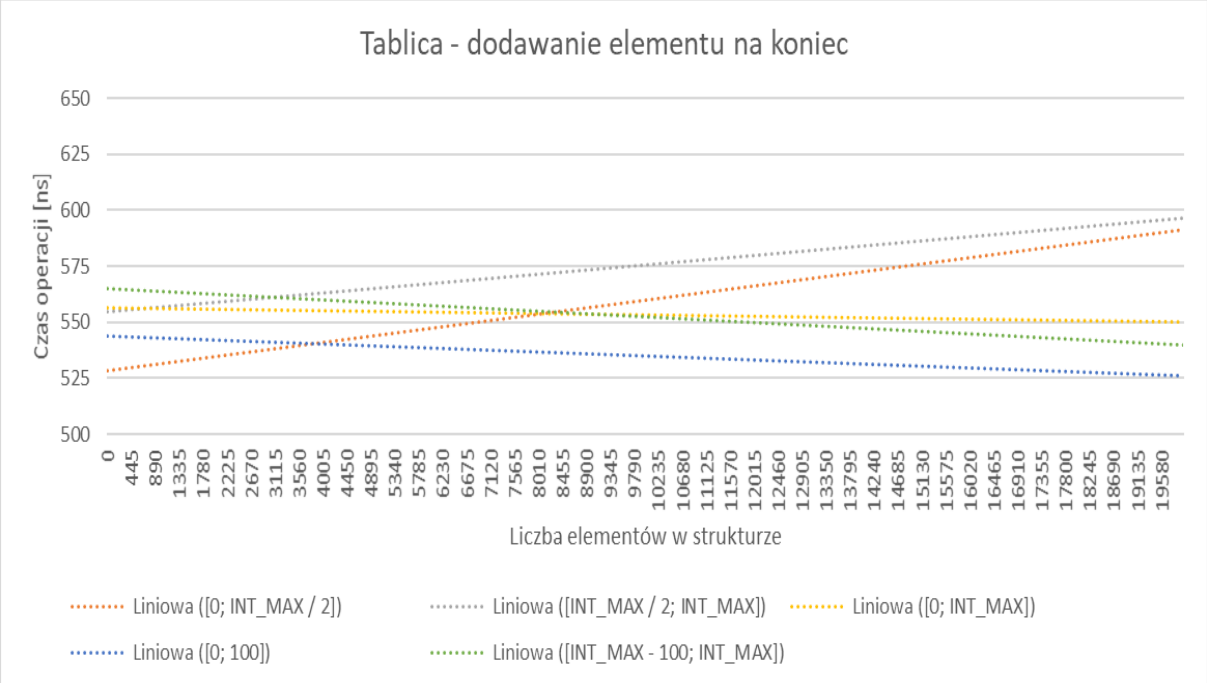
3.1.2. Wstawianie elementu na koniec

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	1258	1020	1024	1341	996
1000	504	677	499	507	691
2000	504	499	499	510	507
3000	522	501	502	502	503
4000	500	497	492	497	490
5000	502	494	506	501	497
6000	501	500	490	650	493
7000	496	499	510	696	501
8000	491	510	508	502	501
9000	501	499	494	497	496
10000	505	509	500	494	493
11000	498	496	933	498	497
12000	497	504	790	497	492
13000	492	664	912	493	496
14000	508	504	503	506	499
15000	507	659	497	496	484
16000	511	504	496	496	500
17000	514	497	699	502	690
18000	503	498	500	500	503
19999	510	830	495	494	490

Tabela 2 Czas [ns] wstawiania elementu na koniec tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 4 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu na koniec tablicy



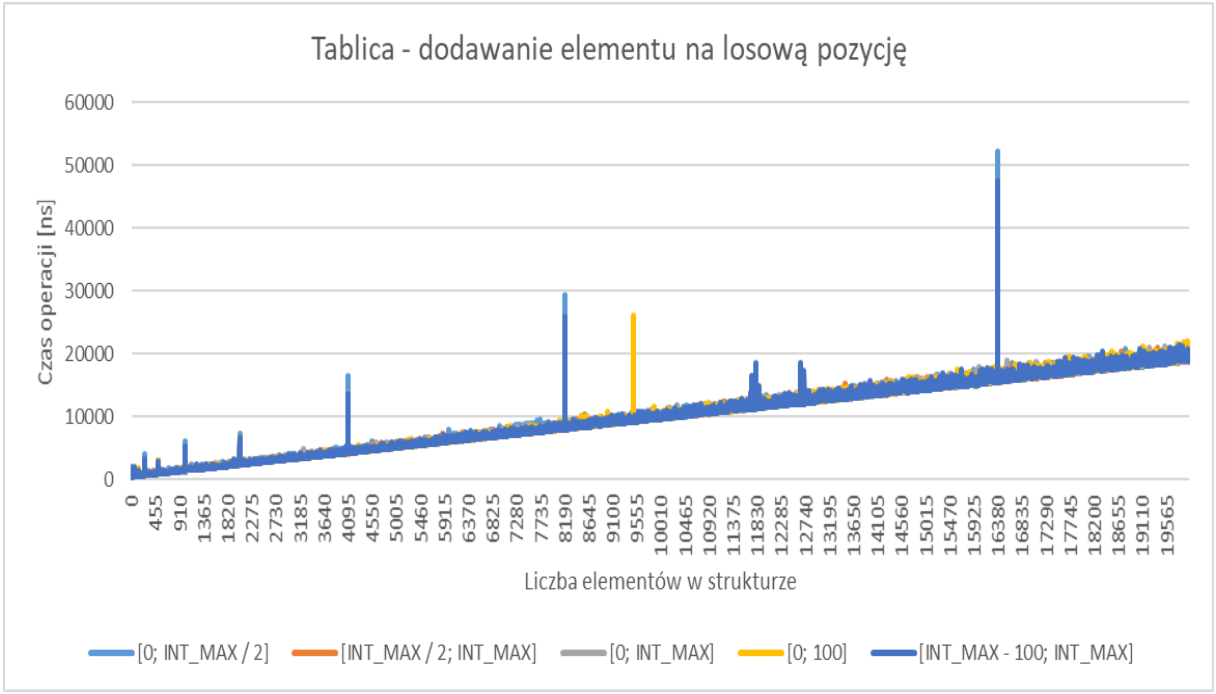
Rysunek 3 Linie trendu dla operacji wstawiania elementu na koniec tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

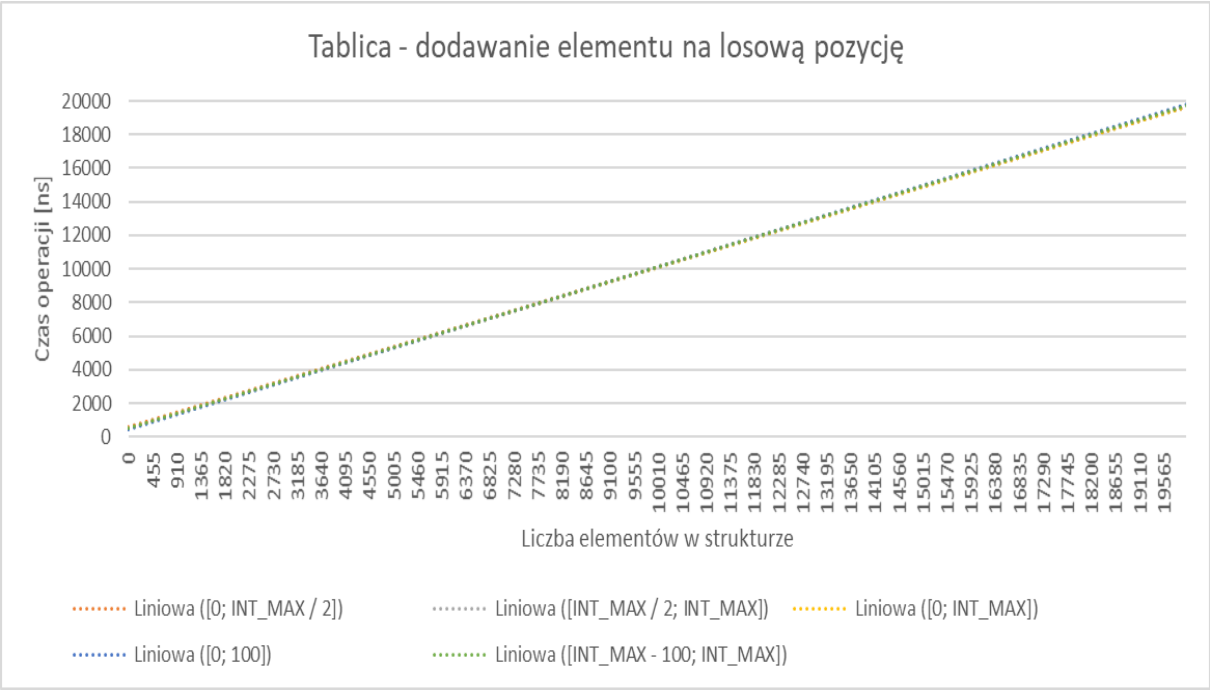
3.1.3. Wstawianie elementu na losową pozycję

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	971	1069	886	1043	896
1000	1417	1548	1404	1547	1867
2000	2321	2310	2305	2314	2340
3000	3510	3237	3368	3239	3622
4000	4182	4308	4137	4135	4494
5000	5102	5421	5224	5399	5552
6000	6049	6028	6011	6316	6942
7000	7213	7630	7211	7507	7334
8000	8514	7966	8280	8019	9052
9000	9754	9441	9214	9587	9294
10000	10642	9798	9844	9689	9668
11000	10744	10798	11904	10949	11339
12000	11844	12365	11783	11847	11794
13000	12544	12918	12886	13490	12968
14000	14570	13650	13312	14554	13821
15000	15010	14926	15071	15450	14444
16000	15747	15521	16179	16606	15396
17000	17011	16439	17122	16929	16659
18000	17527	18108	18470	17162	17658
19999	18876	19558	20024	19124	19673

Tabela 3 Czas [ns] wstawiania elementu w losowe miejsce tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 5 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu w losowe miejsce tablicy



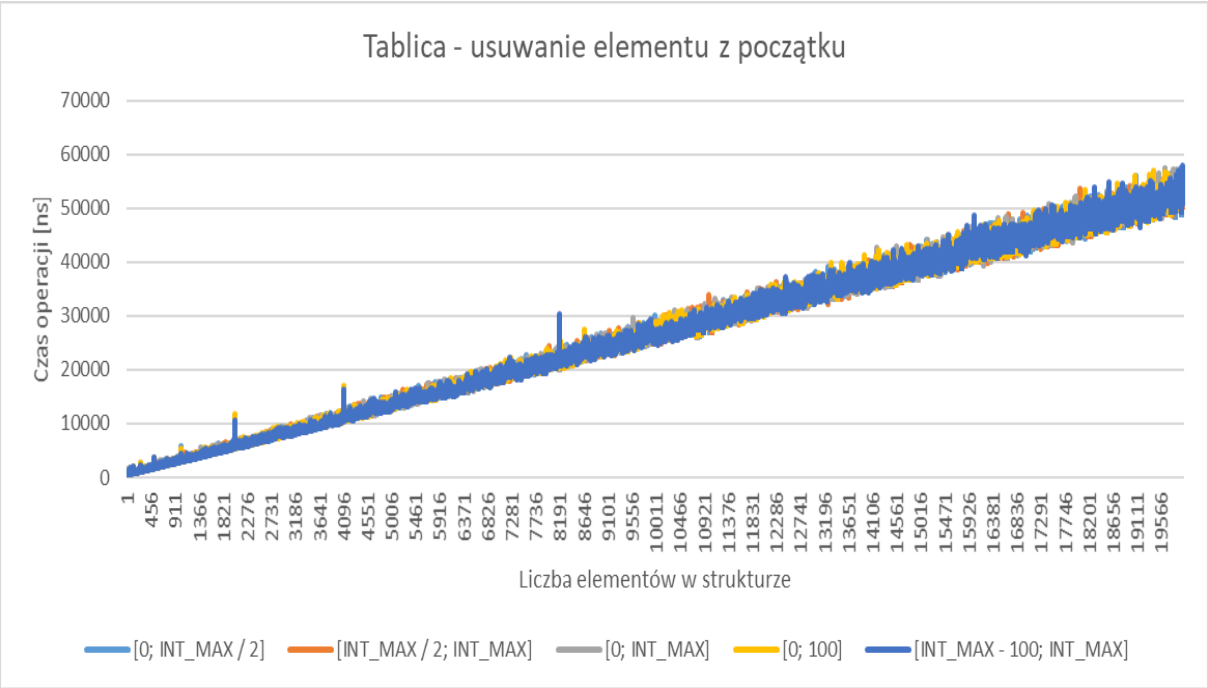
Rysunek 6 Linie trendu dla operacji wstawiania elementu w losowe miejsce tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

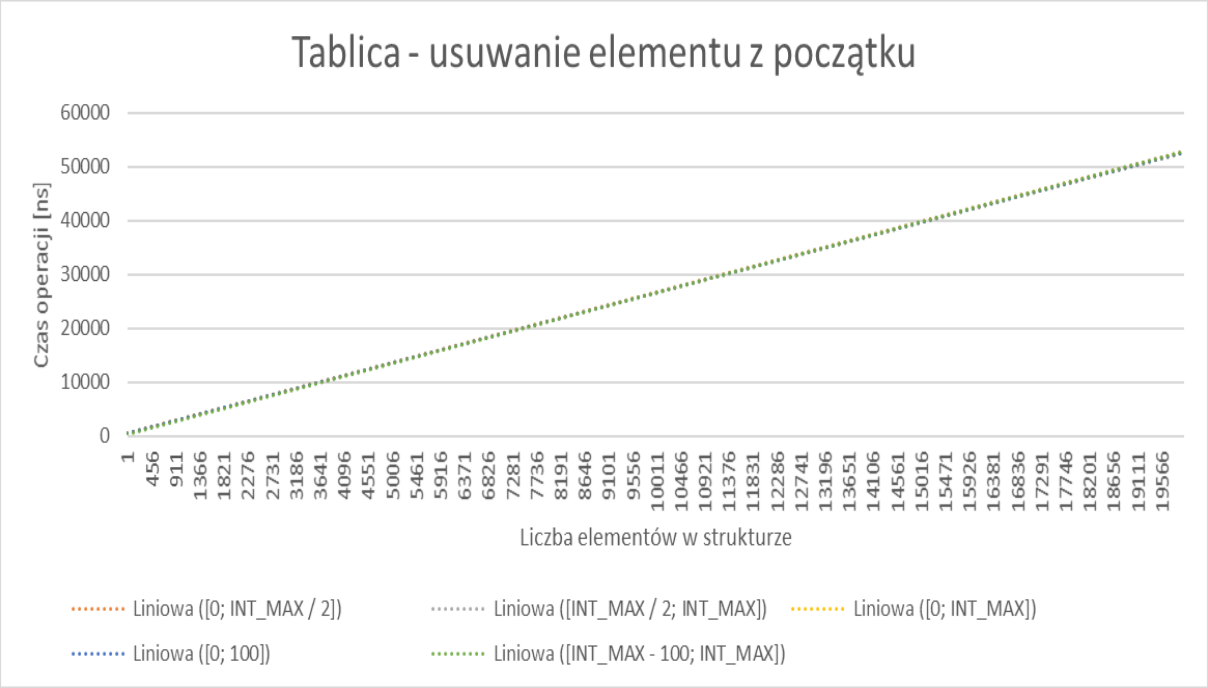
3.1.4. Usuwanie elementu z początku

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	684	673	542	544	550
1000	2861	3089	2879	3066	2860
2000	5903	5699	5754	5478	6157
3000	8190	7952	8187	8560	7814
4000	11162	10899	11095	10578	10846
5000	13250	13277	13399	13669	13319
6000	17527	16304	16904	15495	16680
7000	18811	18601	18516	19984	19075
8000	20397	20998	21045	20713	21599
9000	25005	24213	25045	23397	23479
10000	27697	25758	27313	26511	24820
11000	29128	31173	29712	30562	29276
12000	31409	32968	32013	33497	31711
13000	36705	34012	34626	34708	33208
14000	38617	36767	39601	37187	36708
15000	40083	37921	38819	41717	38478
16000	42698	41900	41488	42119	41835
17000	46543	46905	44904	45527	44062
18000	47290	47407	48699	49838	46371
19999	55983	52148	50973	53755	51774

Tabela 4 Czas [ns] usuwania elementu z początku tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 8 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z początku tablicy



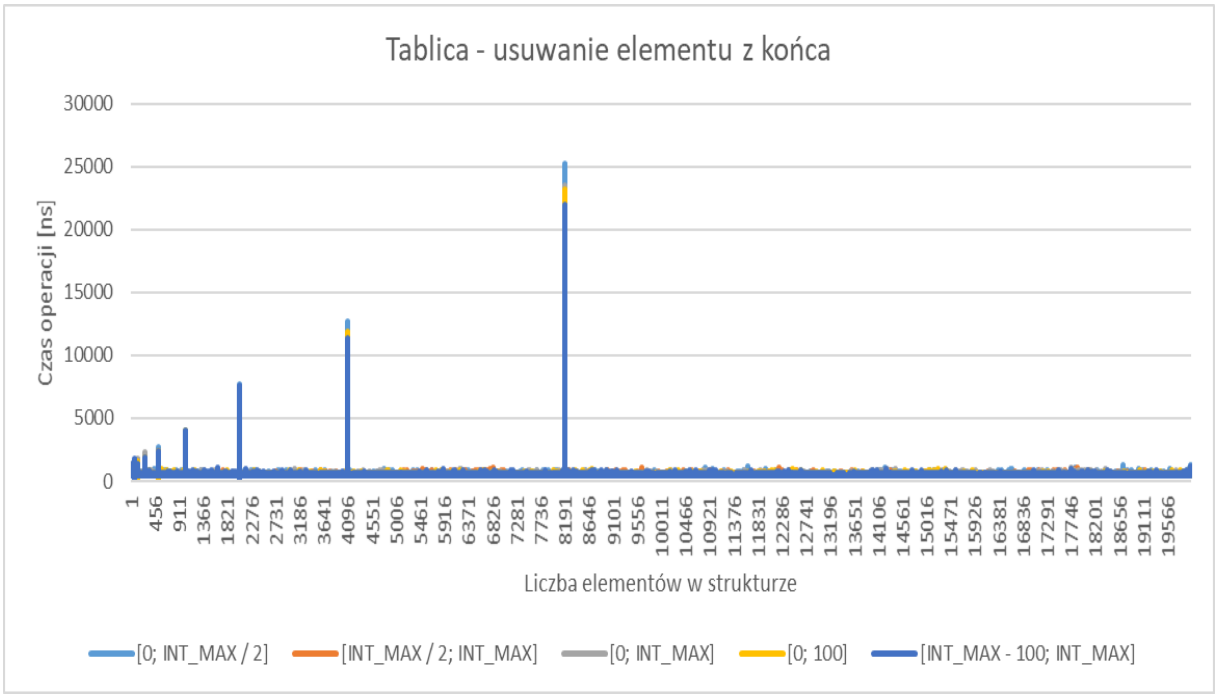
Rysunek 7 Linie trendu dla operacji usuwania elementu z początku tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

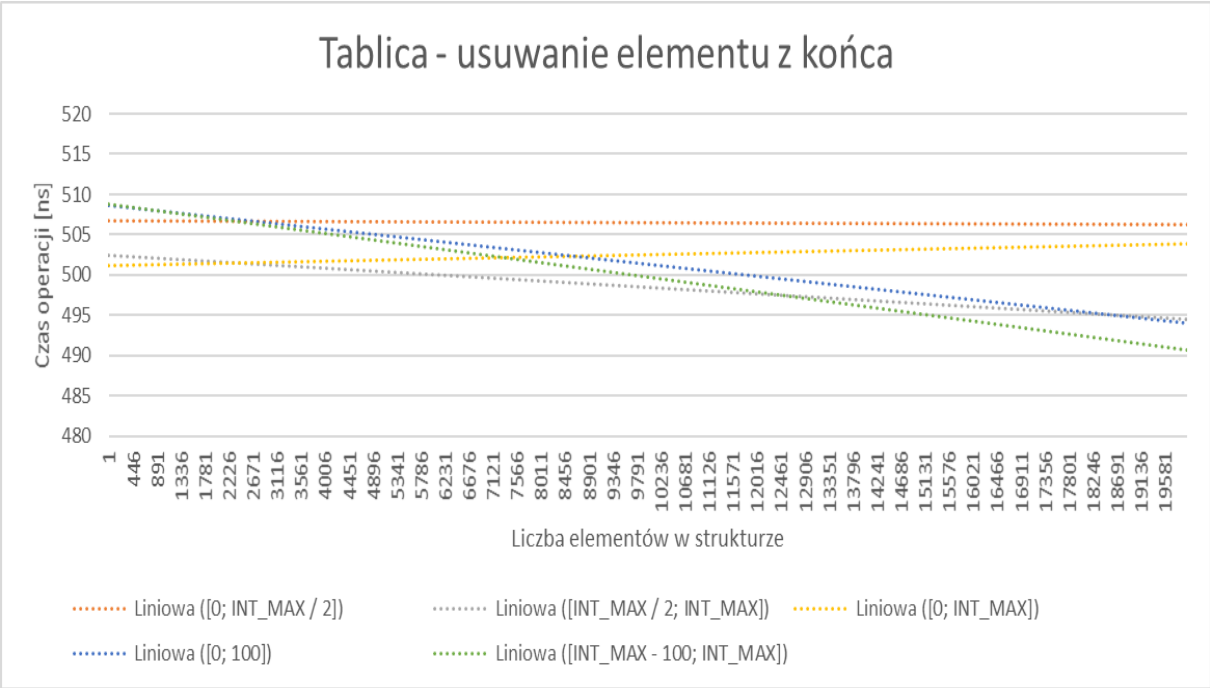
3.1.5. Usuwanie elementu z końca

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	527	529	534	744	534
1000	480	483	473	477	483
2000	476	466	666	479	481
3000	474	477	476	483	477
4000	485	477	478	479	477
5000	475	482	481	475	674
6000	482	476	481	479	482
7000	482	482	479	608	481
8000	483	469	663	483	481
9000	482	480	473	477	485
10000	477	637	482	483	619
11000	475	466	478	480	480
12000	482	664	483	489	474
13000	485	475	479	474	471
14000	476	489	635	477	472
15000	477	479	479	476	477
16000	484	478	480	482	474
17000	479	474	473	478	474
18000	480	478	480	477	479
19999	498	494	500	489	494

Tabela 5 Czas [ns] usuwania elementu z końca tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 9 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z końca tablicy



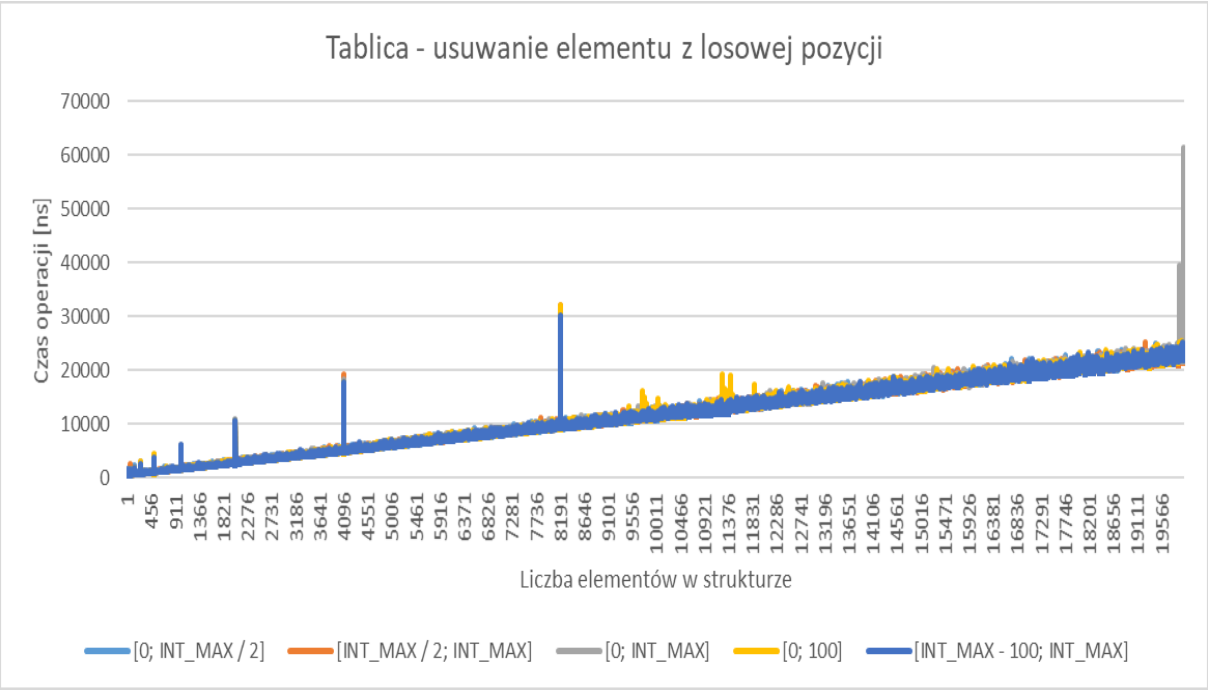
Rysunek 10 Linie trendu dla operacji usuwania elementu z końca tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

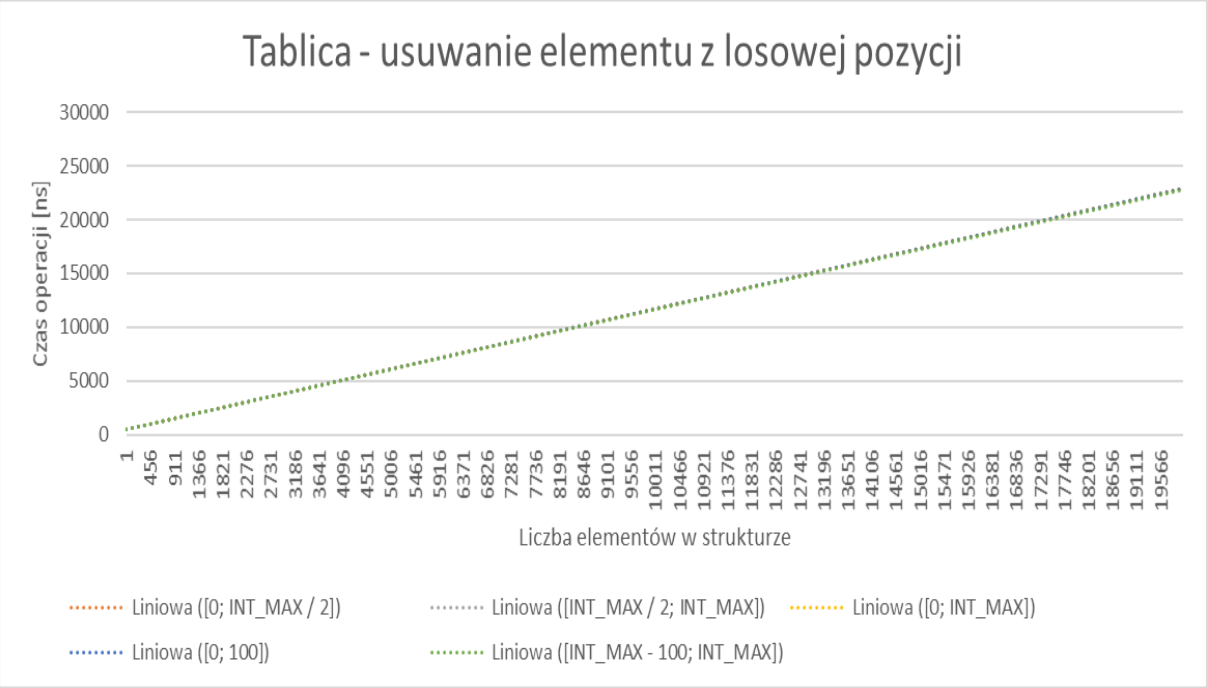
3.1.6. Usuwanie elementu z losowej pozycji

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	562	565	550	564	560
1000	1538	1733	1564	1575	1690
2000	2737	2796	2934	2622	3172
3000	3813	3748	4096	3799	4105
4000	5231	4956	4981	4813	4854
5000	6475	5943	5974	6215	6730
6000	7108	7166	7052	7400	7711
7000	8383	8012	9053	9027	8181
8000	9837	9933	9871	9827	9333
9000	10260	10508	10682	10433	10527
10000	12169	11502	11778	11243	11344
11000	12926	13380	12299	12638	12676
12000	13356	13483	14334	14084	14550
13000	15369	15687	14989	15730	15310
14000	16378	15165	16263	15836	15831
15000	17746	16997	17090	17009	18091
16000	18467	18375	18517	19244	17940
17000	19166	20188	20086	18917	18950
18000	20345	20569	22747	21056	21724
19999	22407	22856	23408	22489	22591

Tabela 6 Czas [ns] usuwania elementu z losowego miejsce tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 11 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z losowego miejsca w tablicy



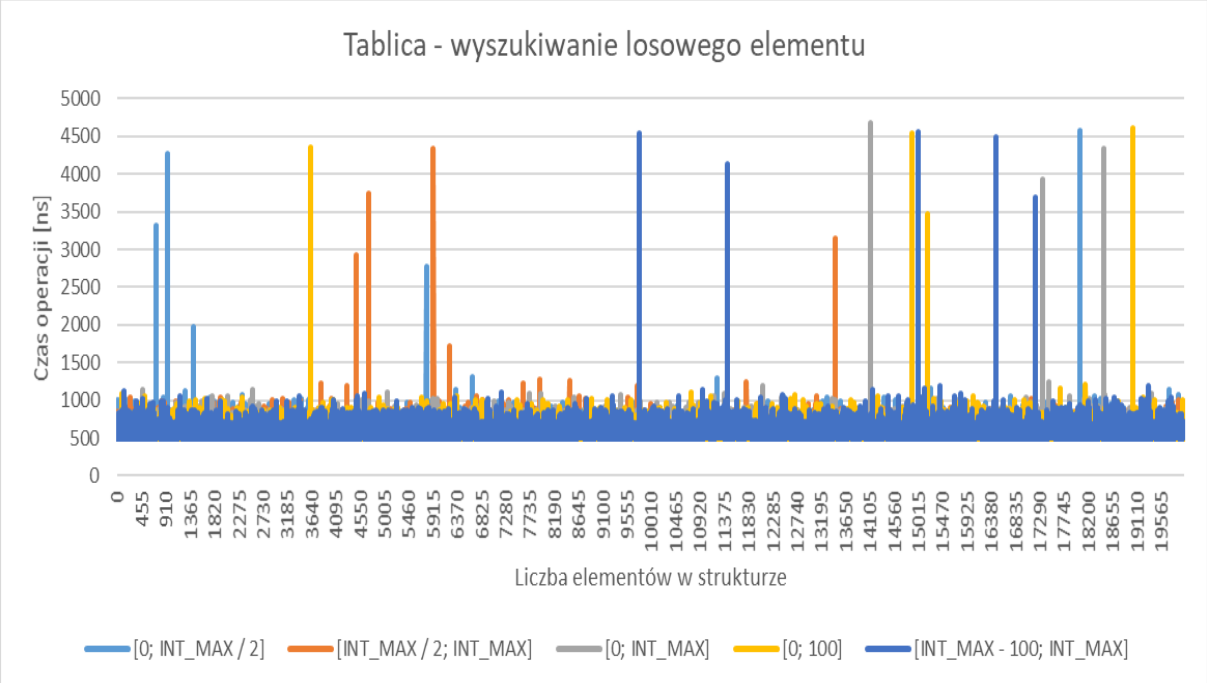
Rysunek 12 Linie trendu dla operacji usuwania elementu z losowego miejsca w tablicy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

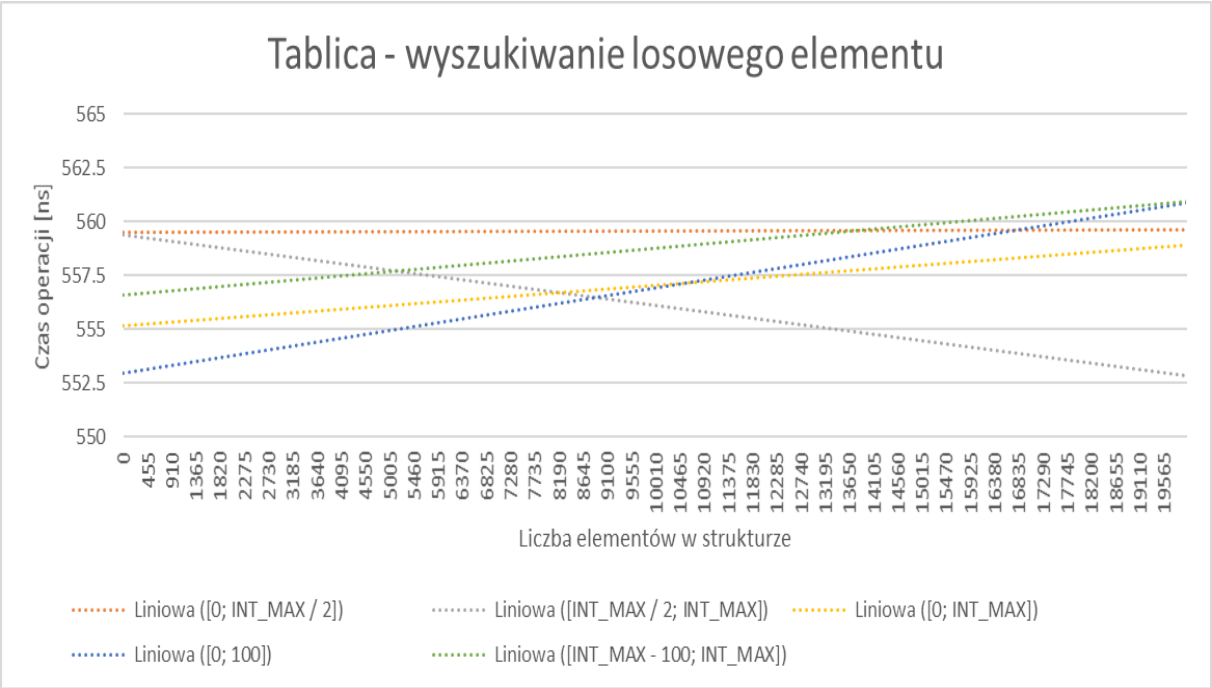
3.1.7. Wyszukiwanie elementu

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	538	514	514	514	515
1000	538	518	684	521	513
2000	518	513	513	510	506
3000	512	644	514	511	520
4000	507	693	508	509	509
5000	518	517	511	697	822
6000	509	513	510	515	510
7000	521	505	511	519	514
8000	516	662	508	699	508
9000	511	516	512	516	711
10000	516	663	513	517	654
11000	516	650	699	518	514
12000	516	715	515	508	510
13000	519	516	707	512	510
14000	659	511	679	554	514
15000	513	509	510	674	508
16000	789	695	515	513	657
17000	512	516	514	747	651
18000	519	670	693	513	512
19999	518	507	517	514	652

Tabela 7 Czas [ns] wyszukiwania elementu w tablicy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 13 Wykres zawierający wszystkie punkty pomiarowe dla operacji wyszukiwania elementu w tablicy



Rysunek 14 Linie trendu dla operacji wyszukiwania elementu w tablicy

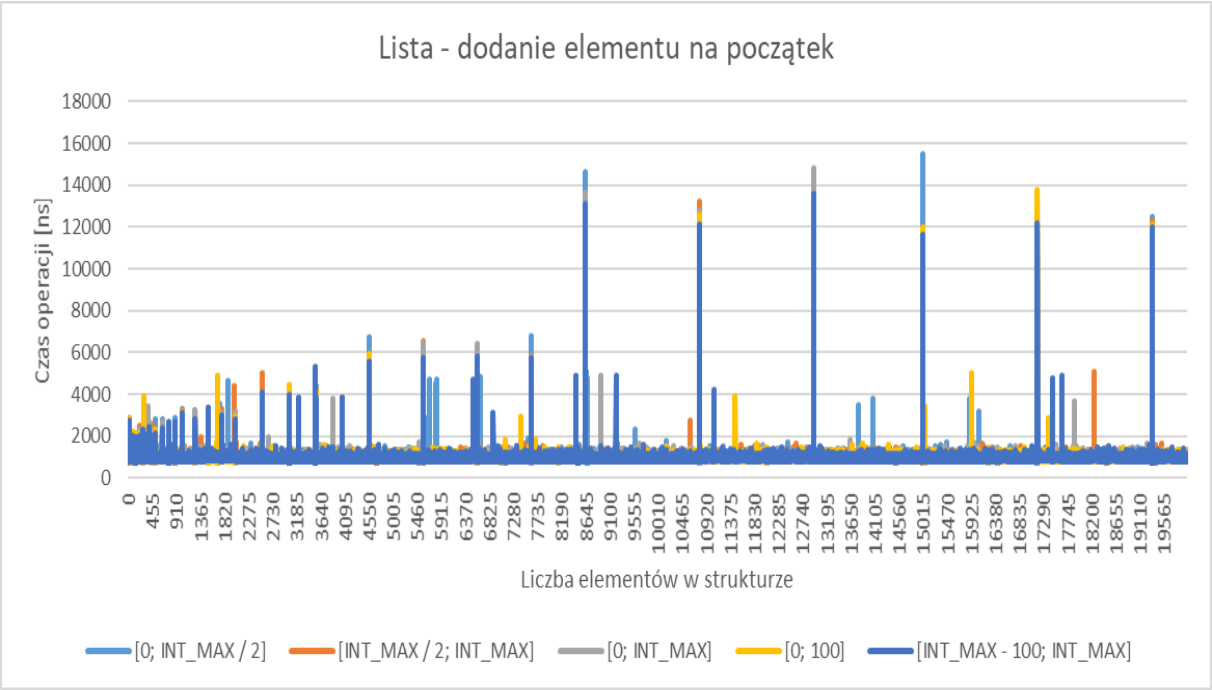
Otrzymane wyniki różnią się od założeń teoretycznych. Czas wyszukiwania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek zastosowanych rozwiązań w architekturze procesora (Intel), które pozwoliły na bardzo wydajny odczyt danych z pamięci (wyszukiwanie nie modyfikuje elementów w strukturze).

3.2. Lista

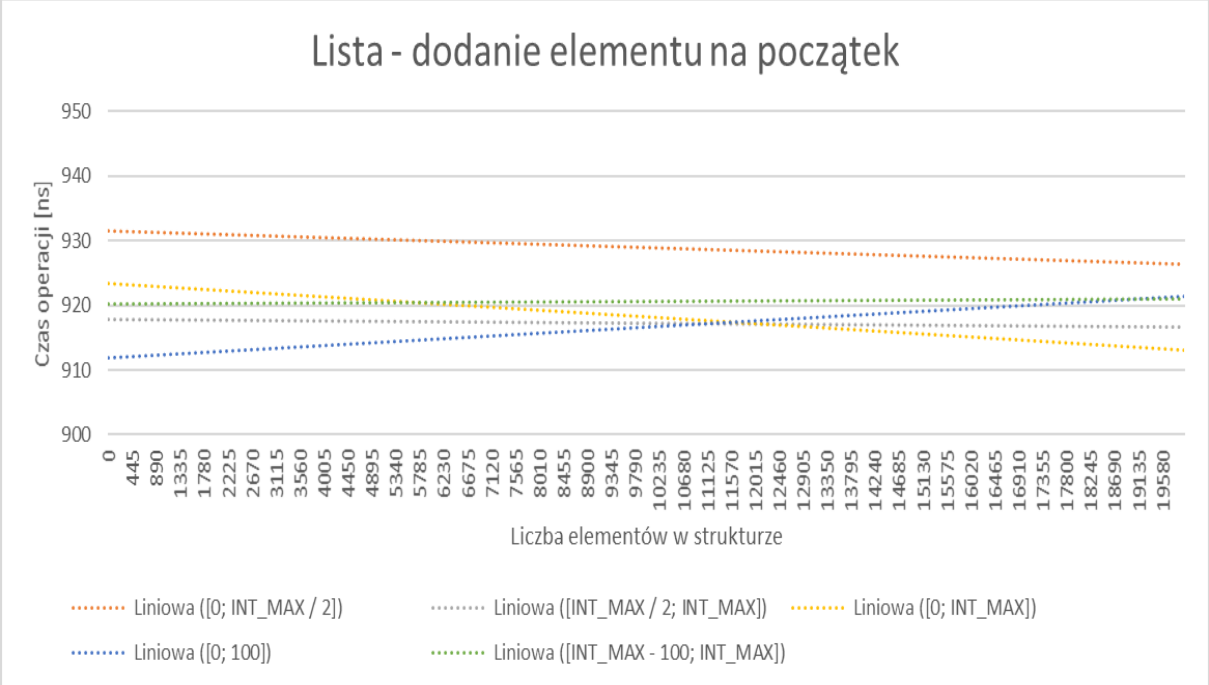
3.2.1. Wstawianie elementu na początek

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	1042	833	833	836	964
1000	854	832	823	819	962
2000	838	838	898	1011	1021
3000	983	953	828	827	952
4000	965	832	836	1016	1014
5000	1030	822	834	958	1155
6000	978	824	841	812	986
7000	1027	825	1181	964	1174
8000	1226	1003	848	956	969
9000	811	862	1151	990	1097
10000	968	839	826	1009	1128
11000	850	830	1026	970	985
12000	818	1167	848	829	978
13000	1135	822	1212	820	843
14000	964	823	819	988	830
15000	1222	1164	855	859	888
16000	1016	977	1014	831	964
17000	840	1021	977	1039	831
18000	860	992	812	824	953
19999	831	830	820	834	823

Tabela 8 Czas [ns] wstawiania elementu na początek listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 15 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu na początek listy



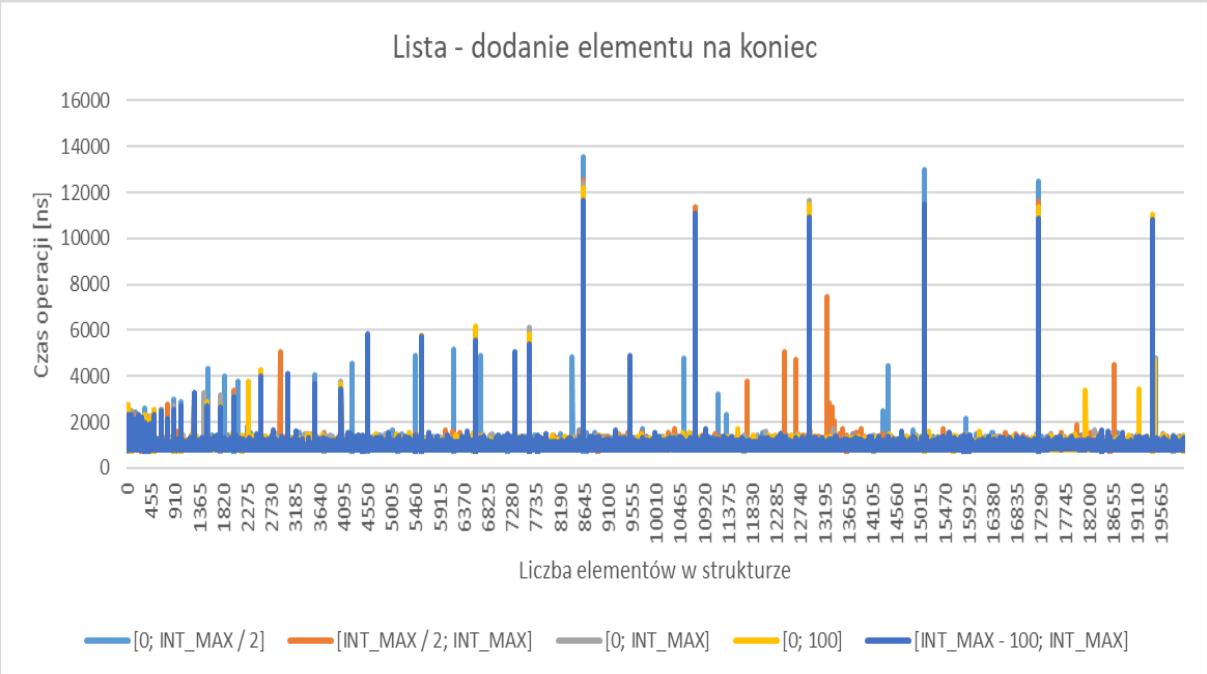
Rysunek 16 Linie trendu dla operacji wstawiania elementu na początek listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

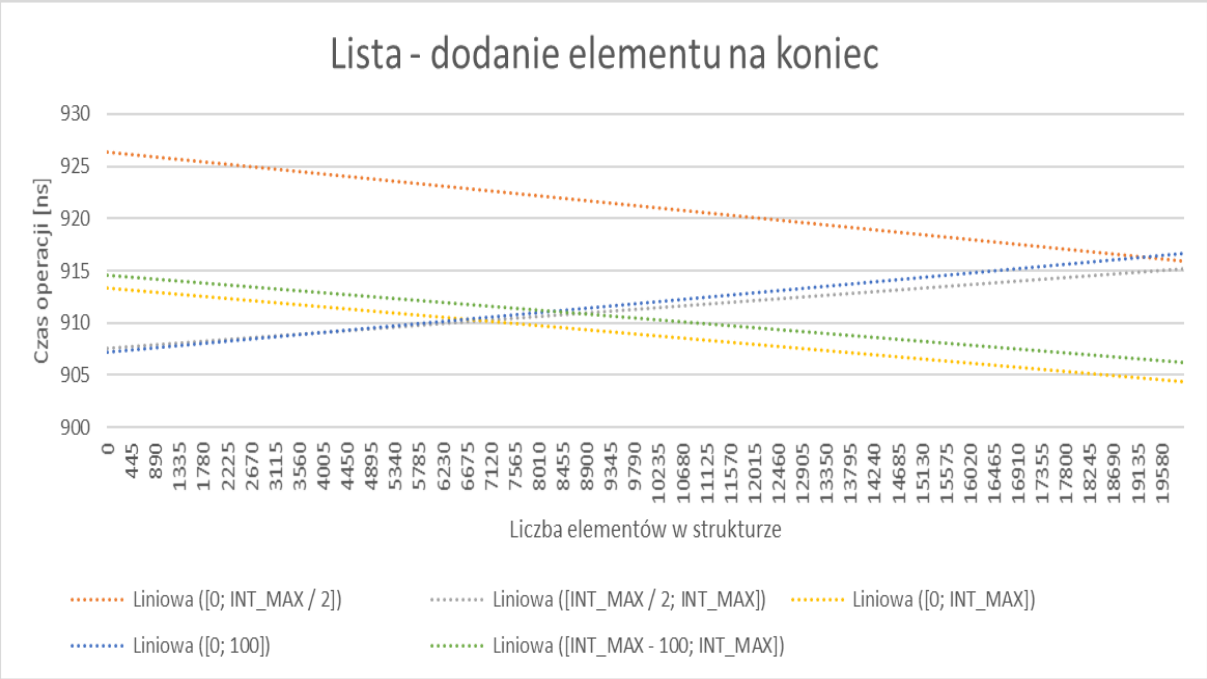
3.2.2. Wstawianie elementu na koniec

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	921	830	821	824	827
1000	838	813	817	1206	836
2000	838	828	1155	829	1137
3000	829	846	826	829	1016
4000	839	1159	829	830	825
5000	823	1033	815	954	815
6000	820	965	824	806	819
7000	831	819	819	836	971
8000	818	812	824	809	956
9000	817	824	853	817	1157
10000	828	955	1229	829	841
11000	961	834	1022	821	817
12000	973	827	819	832	1014
13000	964	1351	1029	829	823
14000	839	964	825	838	975
15000	852	976	825	1310	826
16000	1097	830	836	839	817
17000	1153	826	827	1023	1101
18000	836	998	819	1199	851
19999	822	1105	820	828	837

Tabela 9 Czas [ns] wstawiania elementu na koniec listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 17 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu na koniec listy



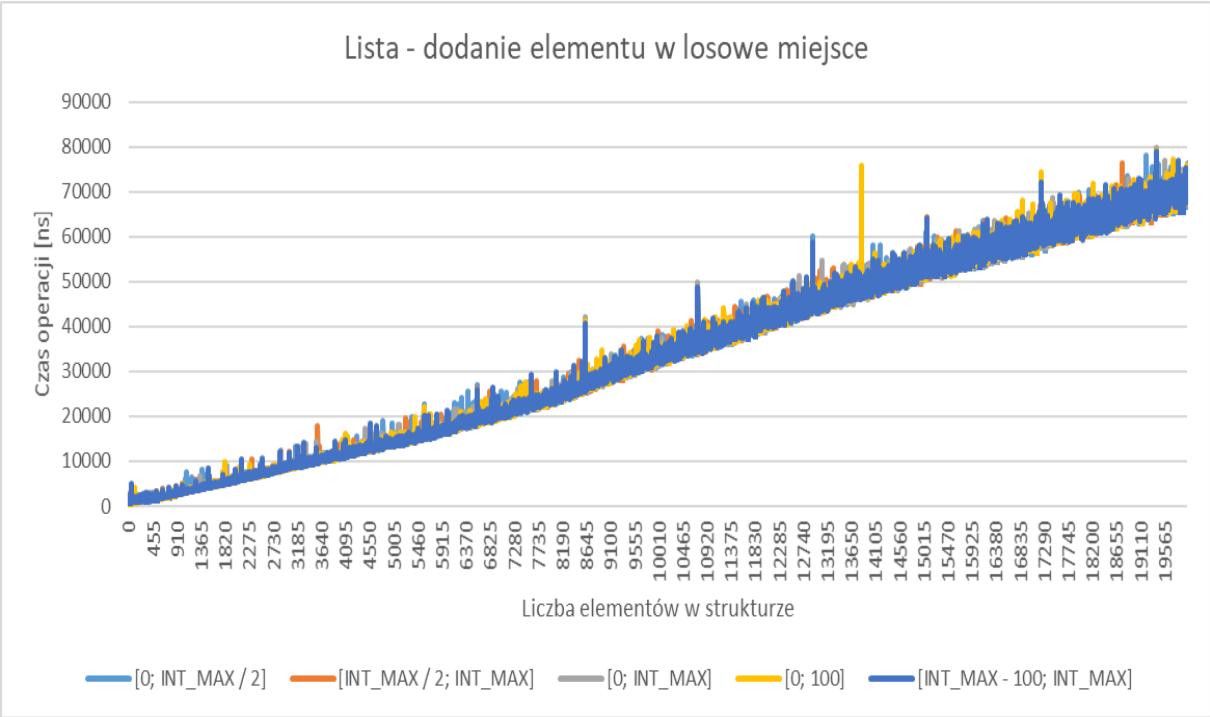
Rysunek 18 Linie trendu dla operacji wstawiania elementu na koniec listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

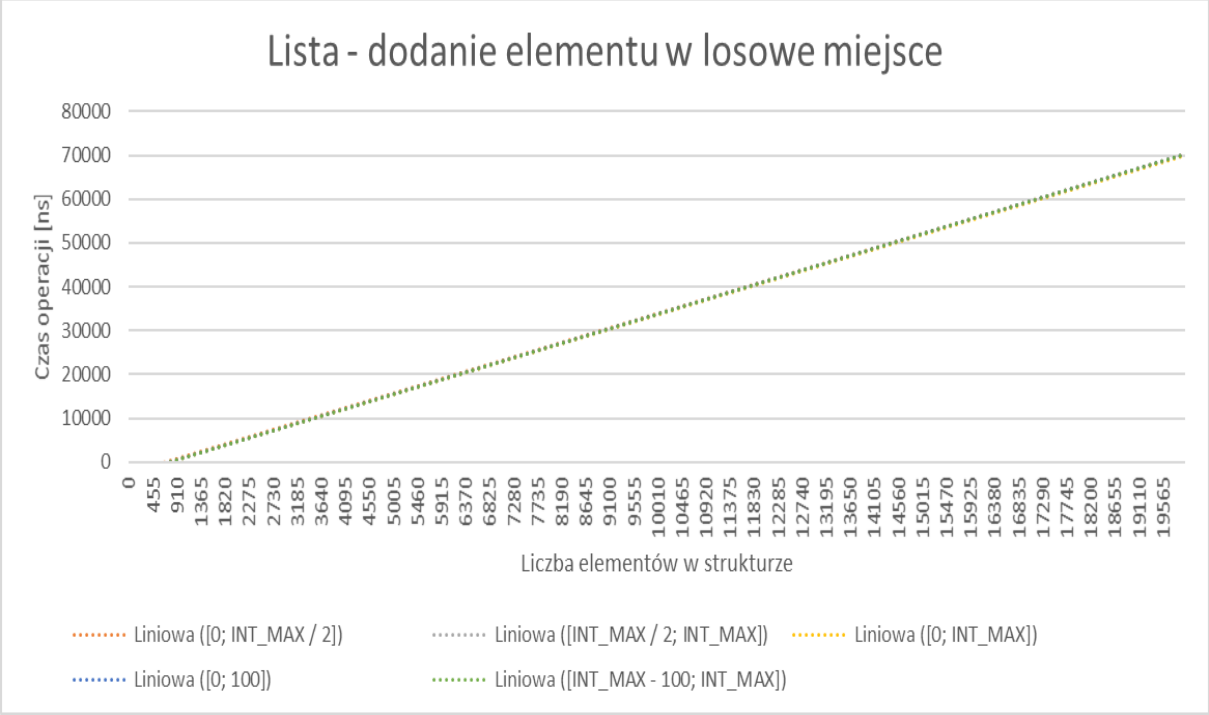
3.2.3. Wstawianie elementu na losową pozycję

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	839	822	827	800	1002
1000	2891	2858	2897	3247	3010
2000	5384	5771	5336	5391	6228
3000	8681	8836	8398	9043	8563
4000	11449	11710	11215	11222	10986
5000	13483	14679	14892	14402	14045
6000	18394	16773	17497	16864	17054
7000	21265	20900	20265	20932	20766
8000	23242	23197	25706	24268	24184
9000	28767	28748	28640	28791	28902
10000	33470	32279	32153	32635	33497
11000	36980	38404	36736	37863	37297
12000	41909	41140	41415	40481	41351
13000	46053	44061	45956	46052	46102
14000	48394	47306	50890	49003	48397
15000	53386	51743	50611	53534	53049
16000	55415	58711	58965	55322	56045
17000	59461	57280	59507	57443	61750
18000	61587	62289	61703	63725	63878
19999	71288	68575	70735	70881	69282

Tabela 10 Czas [ns] wstawiania elementu na losową pozycję listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 19 Wykres zawierający wszystkie punkty pomiarowe dla operacji wstawiania elementu na losową pozycję listy



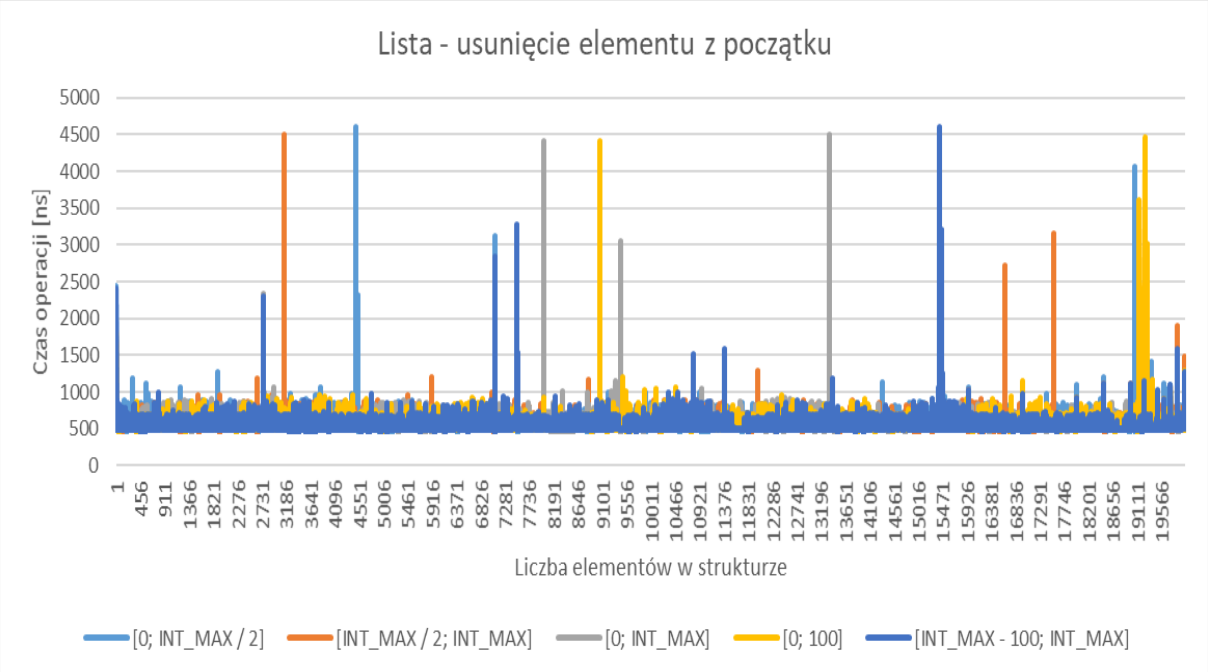
Rysunek 20 Linie trendu dla operacji wstawiania elementu na losową pozycję listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

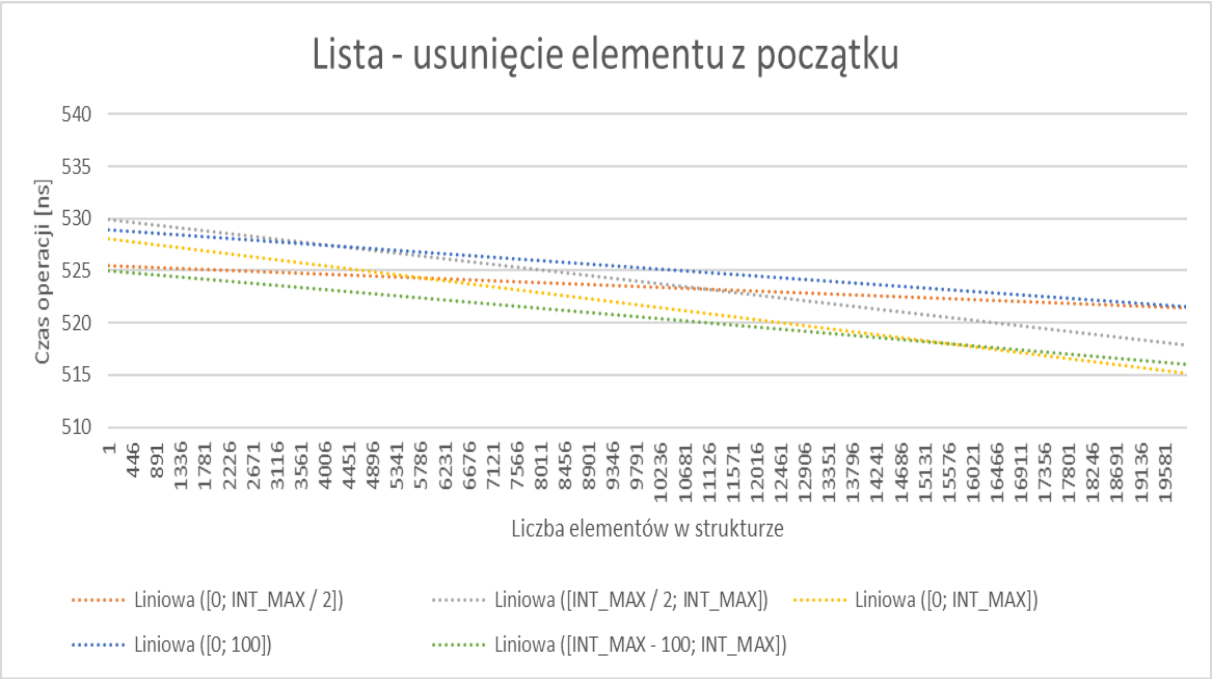
3.2.4. Usuwanie elementu z początku

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	2444	1845	1801	1813	2404
1000	687	635	494	494	495
2000	496	500	497	491	497
3000	498	501	686	497	494
4000	501	500	495	497	494
5000	500	505	660	691	496
6000	496	502	500	628	495
7000	495	508	499	505	500
8000	491	493	498	502	501
9000	499	633	498	498	493
10000	497	500	503	638	500
11000	490	495	501	494	496
12000	492	503	498	507	502
13000	498	501	497	498	501
14000	685	631	495	502	690
15000	506	502	492	492	490
16000	502	497	494	507	709
17000	492	494	499	500	496
18000	498	494	489	495	493
19999	669	526	521	535	521

Tabela 11 Czas [ns] usuwania elementu z początku listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 21 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z początku listy



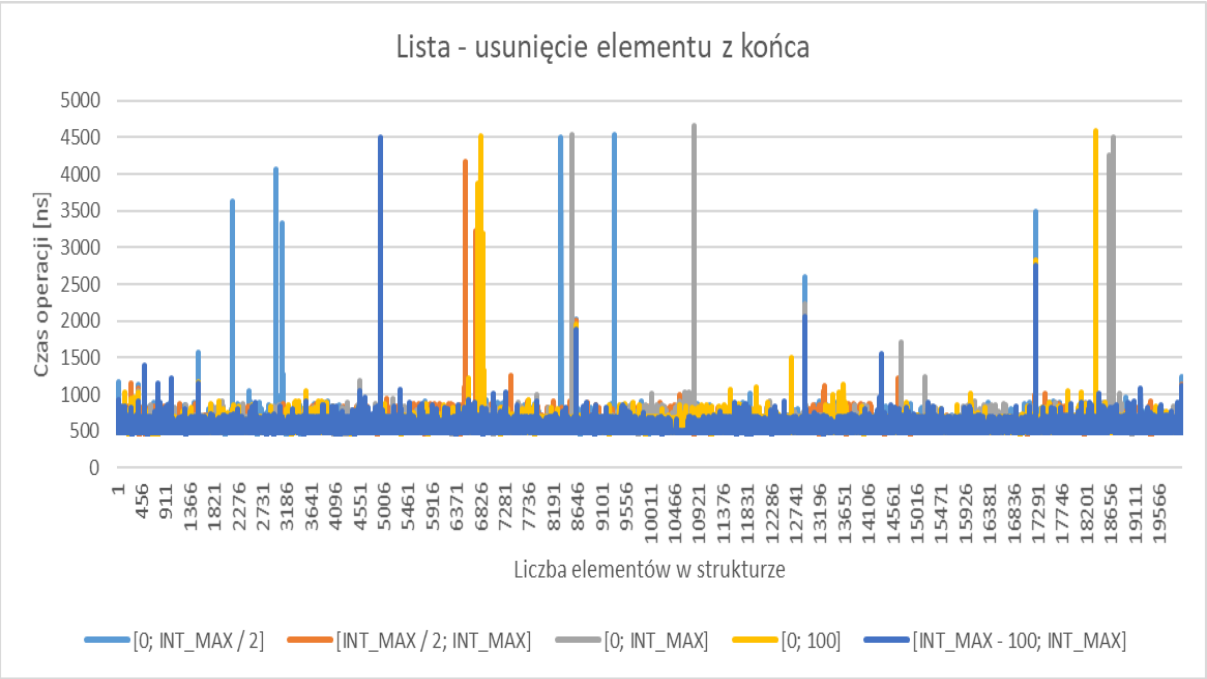
Rysunek 22 Linie trendu dla operacji usuwana z początku listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

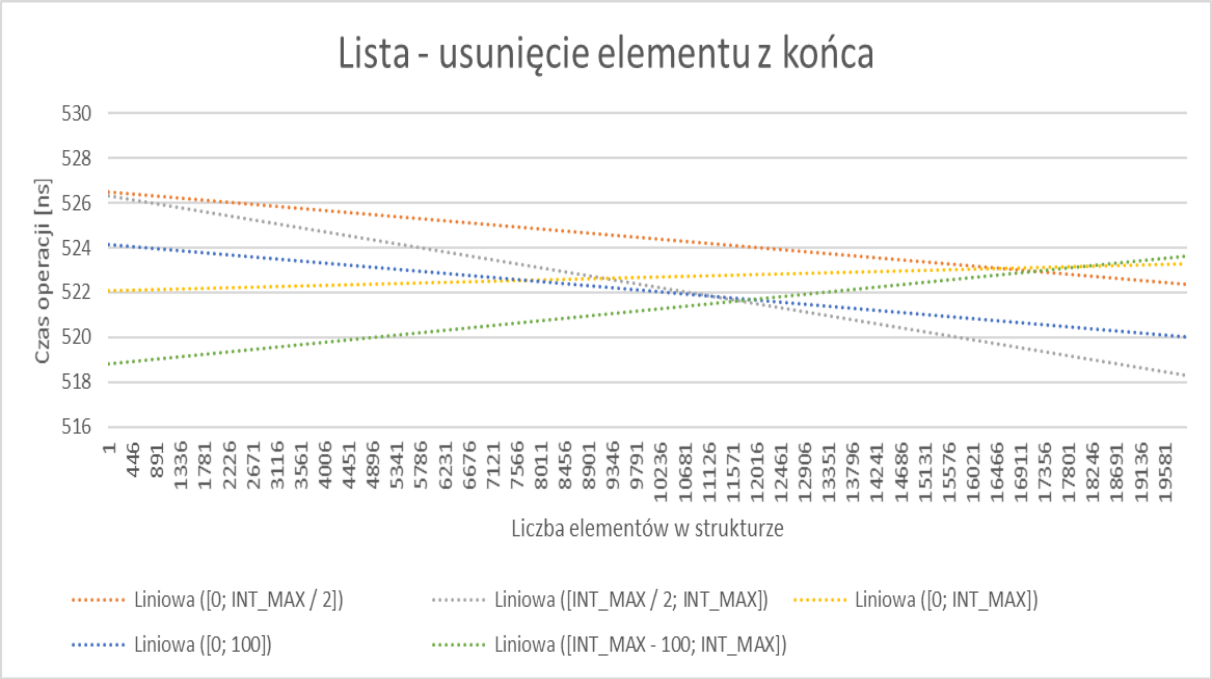
3.2.5. Usuwanie elementu z końca

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	494	494	491	495	498
1000	506	500	502	499	497
2000	503	500	500	503	500
3000	690	497	500	504	500
4000	502	499	500	503	494
5000	499	502	663	499	696
6000	496	504	496	499	498
7000	636	500	498	501	499
8000	505	497	497	496	499
9000	505	668	498	502	499
10000	497	499	501	490	499
11000	499	737	504	502	494
12000	500	493	503	500	493
13000	498	682	499	636	631
14000	507	500	510	503	506
15000	499	495	500	496	496
16000	662	495	493	499	495
17000	500	692	497	500	501
18000	505	505	500	500	501
19999	520	526	525	522	526

Tabela 12 Czas [ns] usuwania elementu z końca listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 24 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z końca listy



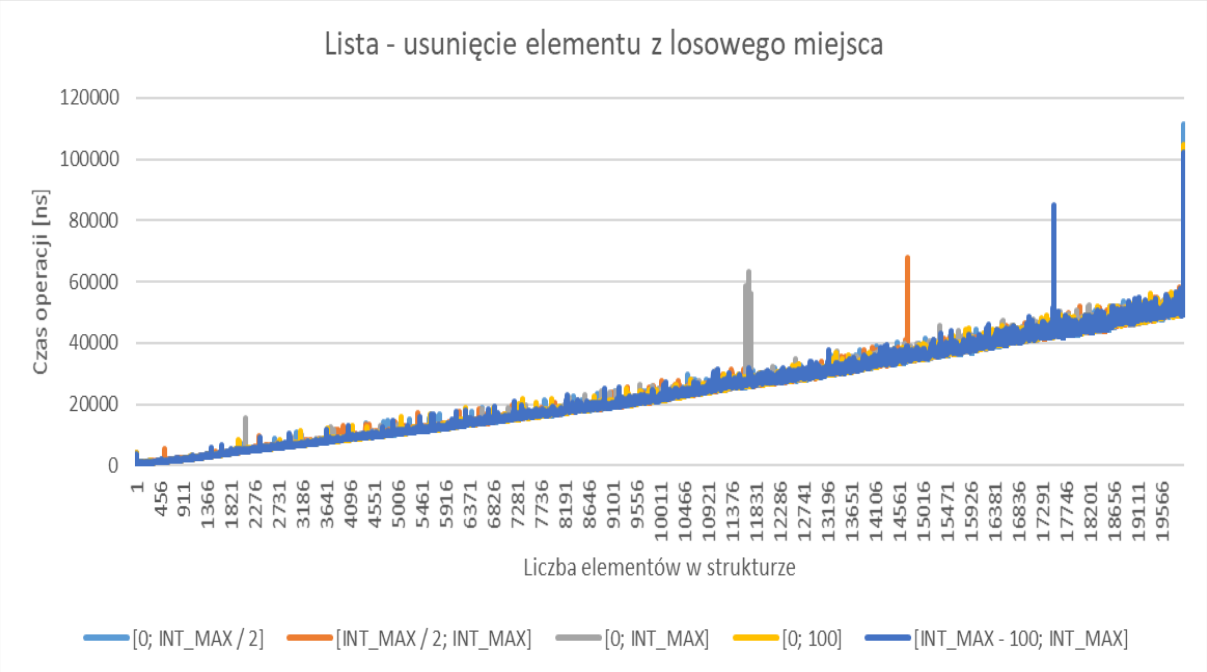
Rysunek 23 Linie trendu dla operacji usuwana z końca listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

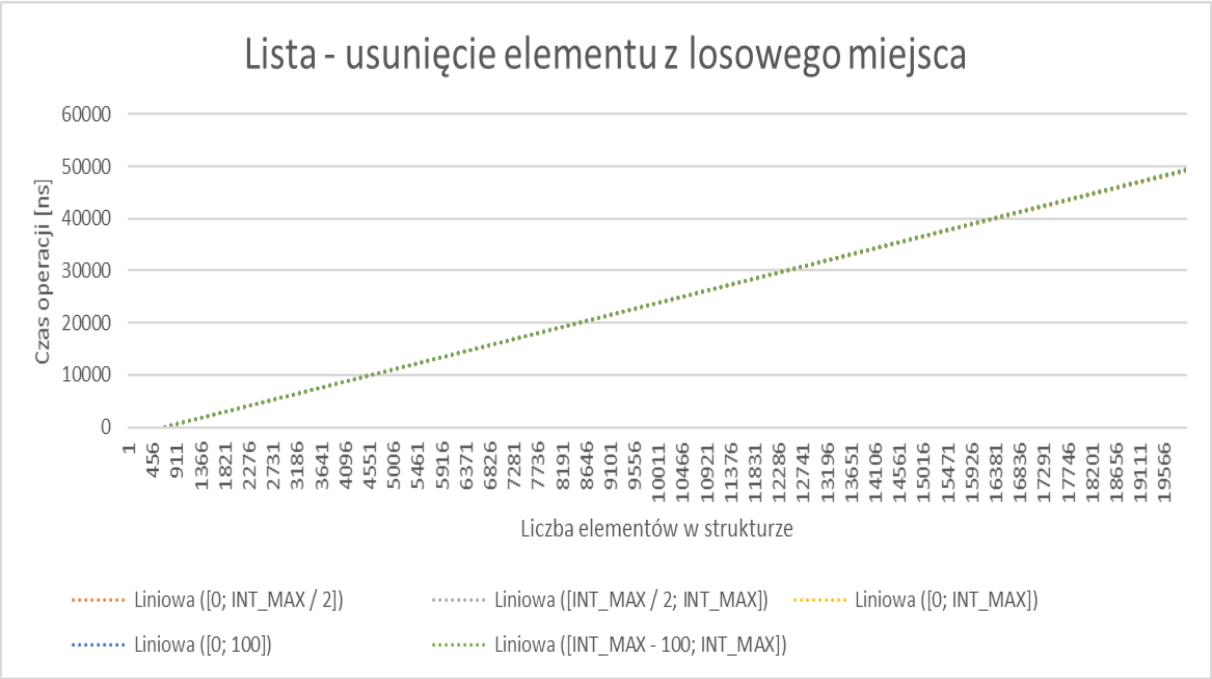
3.2.6. Usuwanie elementu z losowej pozycji

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	724	728	543	539	542
1000	2055	2054	2062	2080	2196
2000	4820	4429	4575	4597	4608
3000	7057	6961	6703	6613	6928
4000	8882	9158	8697	9048	8674
5000	11458	10744	10964	11449	10703
6000	12969	13103	12768	12504	13125
7000	14751	14930	15246	15622	15121
8000	17983	17067	17645	16639	17487
9000	20015	20321	20780	19286	19657
10000	21439	22099	22125	22165	21244
11000	25451	25645	25172	24864	25073
12000	27781	29111	28690	27326	28775
13000	30075	29729	31948	30353	30715
14000	33320	32789	33556	32204	32389
15000	36156	36552	35800	36142	36263
16000	38985	39077	39187	38846	39158
17000	44288	43298	40703	44305	45018
18000	43927	46959	47116	46461	44120
19999	56309	54904	54923	54487	55001

Tabela 13 Czas [ns] usuwania elementu z losowej pozycji listy w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 25 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania elementu z losowej pozycji listy



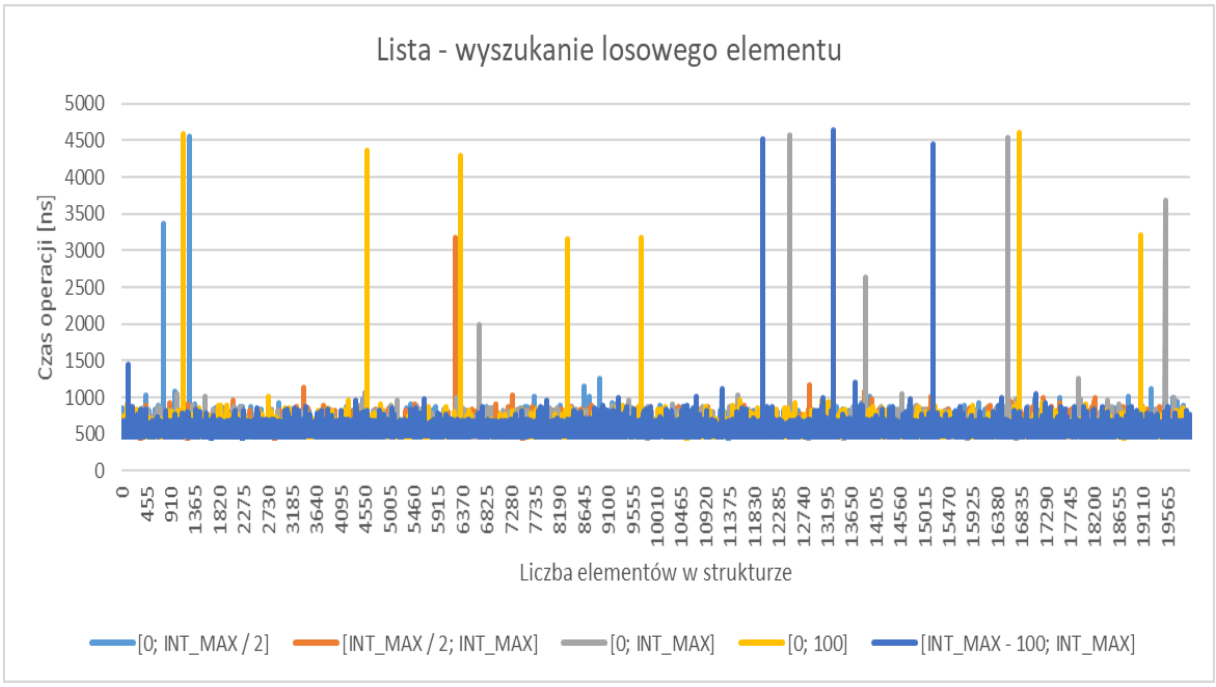
Rysunek 26 Linie trendu dla operacji usuwana z losowej pozycji listy

Otrzymane wyniki są zgodne z założeniami teoretycznymi. Brak zauważalnych różnic dla różnych zakresów kluczy.

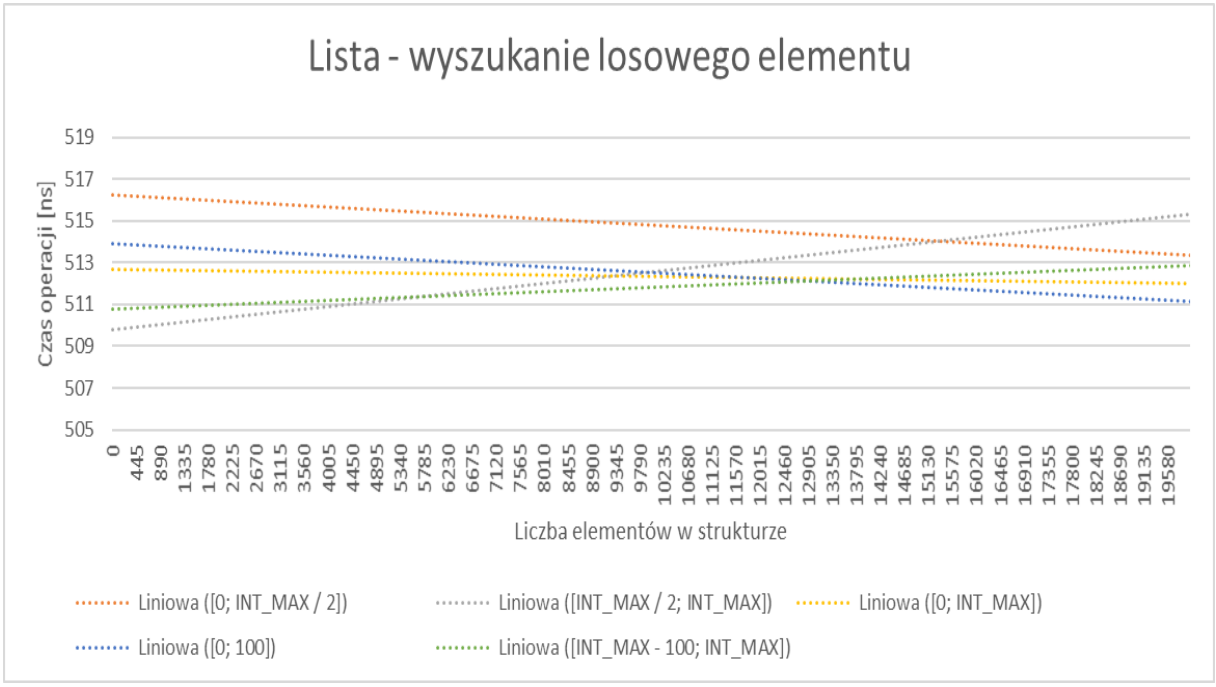
3.2.7. Wyszukiwanie elementu

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	530	502	508	503	502
1000	491	483	485	487	488
2000	481	491	489	619	492
3000	490	675	480	487	487
4000	502	487	483	491	480
5000	486	493	641	480	492
6000	678	644	487	495	487
7000	496	487	491	480	486
8000	494	488	495	487	488
9000	500	491	486	482	482
10000	478	493	488	491	490
11000	492	643	491	485	487
12000	492	491	490	488	493
13000	485	491	485	483	489
14000	682	493	488	828	675
15000	484	486	493	484	626
16000	489	489	486	614	485
17000	491	488	485	482	627
18000	621	493	485	482	485
19999	486	483	497	487	485

Tabela 14 Czas [ns] wyszukiwania elementu na liście w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 28 Wykres zawierający wszystkie punkty pomiarowe dla operacji wyszukiwania elementu na liście



Rysunek 27 Linie trendu dla operacji wyszukiwania elementu na liście

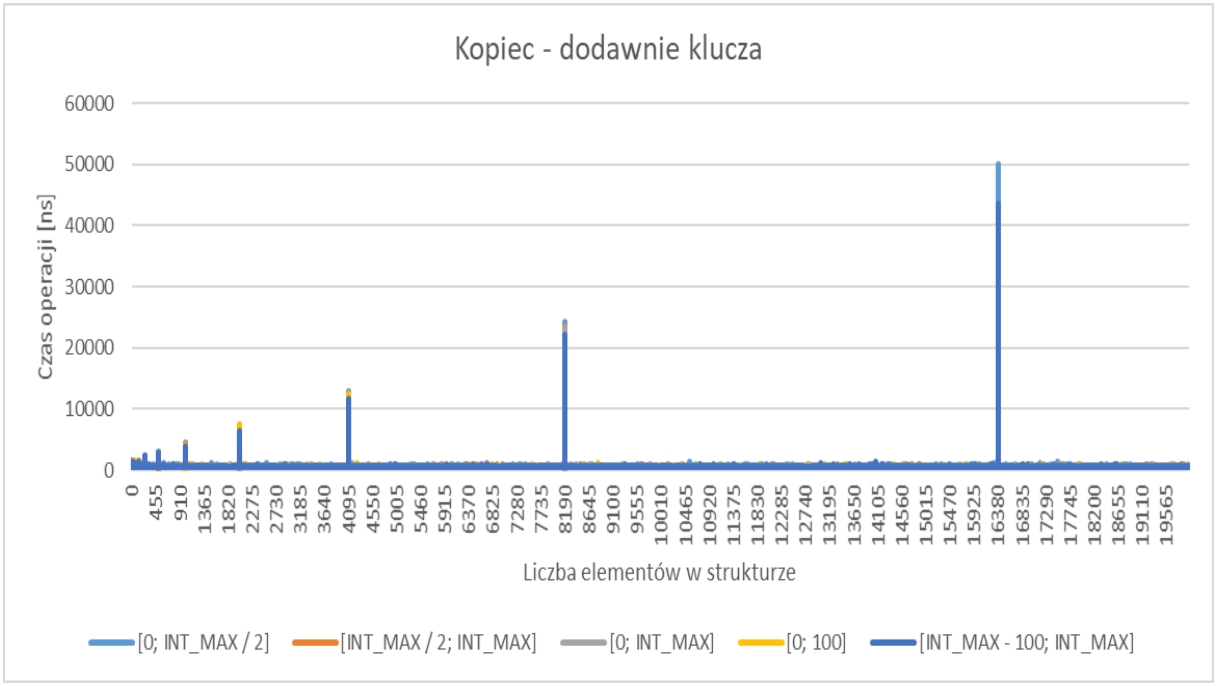
Otrzymane wyniki różnią się od założeń teoretycznych. Czas wyszukiwania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek zastosowanych rozwiązań w architekturze procesora (Intel), które pozwoliły na bardzo wydajny odczyt danych z pamięci (wyszukiwanie nie modyfikuje elementów w strukturze).

3.3. Kopiec

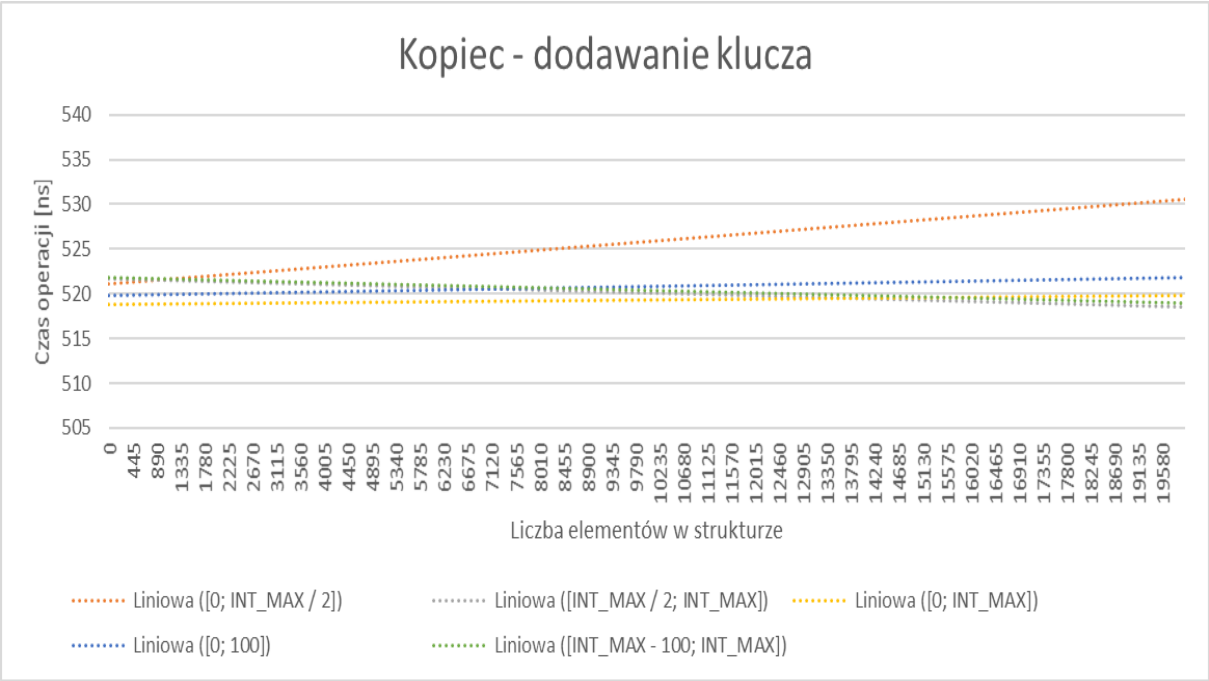
3.3.1. Dodawanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	1003	1130	937	1086	921
1000	502	500	497	493	499
2000	509	492	501	496	501
3000	658	666	489	493	492
4000	494	675	492	499	495
5000	645	491	489	492	501
6000	485	490	623	493	495
7000	489	498	491	491	492
8000	488	499	494	491	490
9000	493	488	495	632	677
10000	624	501	491	493	484
11000	497	479	624	491	623
12000	498	492	491	489	486
13000	483	649	495	640	492
14000	495	493	495	492	633
15000	493	496	491	491	496
16000	836	490	489	494	821
17000	508	686	490	490	482
18000	490	495	492	494	672
19999	492	487	491	492	496

Tabela 15 Czas [ns] dodawania klucza do kopca w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 29 Wykres zawierający wszystkie punkty pomiarowe dla operacji dodawania klucza do kopca



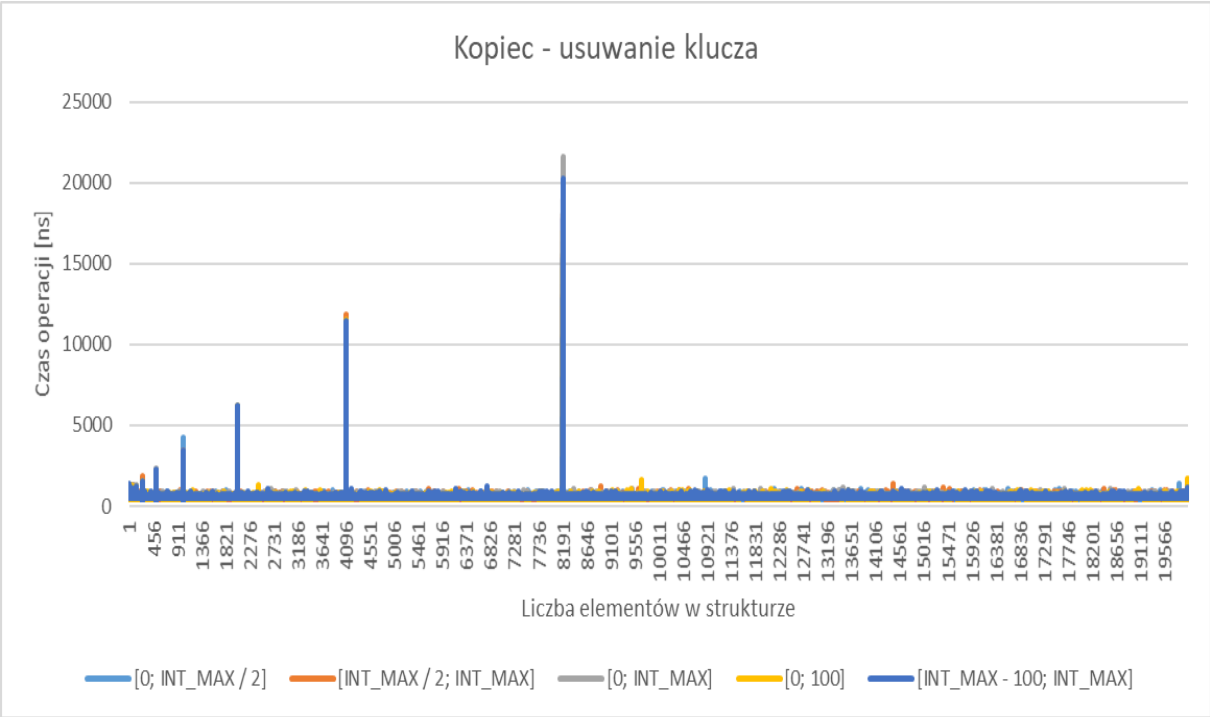
Rysunek 30 Linie trendu dla operacji dodawania klucza do kopca

Otrzymane wyniki różnią się od założeń teoretycznych. Czas dodawania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek powolnego tempa wzrostu czasu wykonywania operacji o złożoności logarytmicznej.

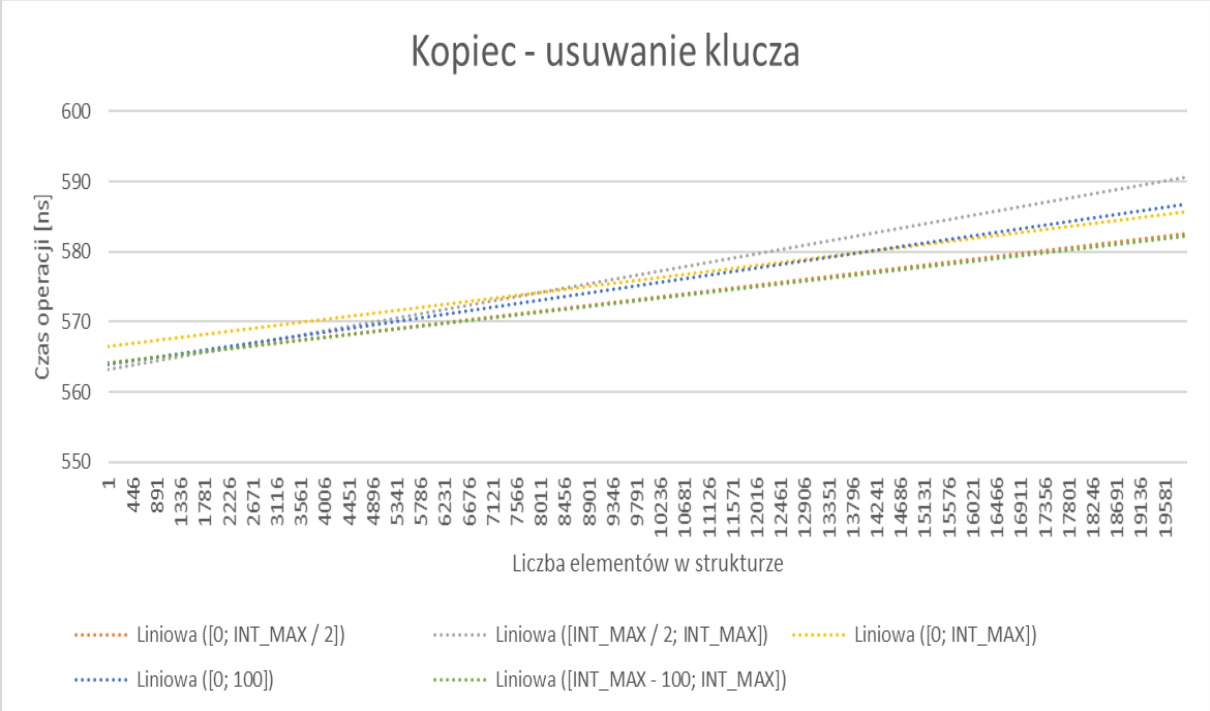
3.3.2. Usuwanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	571	570	721	572	573
1000	528	527	528	534	528
2000	537	524	543	533	532
3000	539	532	534	541	533
4000	732	528	678	527	537
5000	540	663	546	547	539
6000	559	547	545	533	541
7000	542	538	551	542	538
8000	536	537	529	535	538
9000	553	694	541	546	554
10000	736	542	547	708	538
11000	562	549	550	746	548
12000	554	686	708	555	548
13000	547	554	549	563	558
14000	568	555	542	568	545
15000	558	550	565	560	801
16000	544	554	544	555	731
17000	541	543	559	698	548
18000	712	555	713	549	693
19999	617	783	580	593	581

Tabela 16 Czas [ns] usuwania klucza z kopca w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 31 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwania klucza z kopca



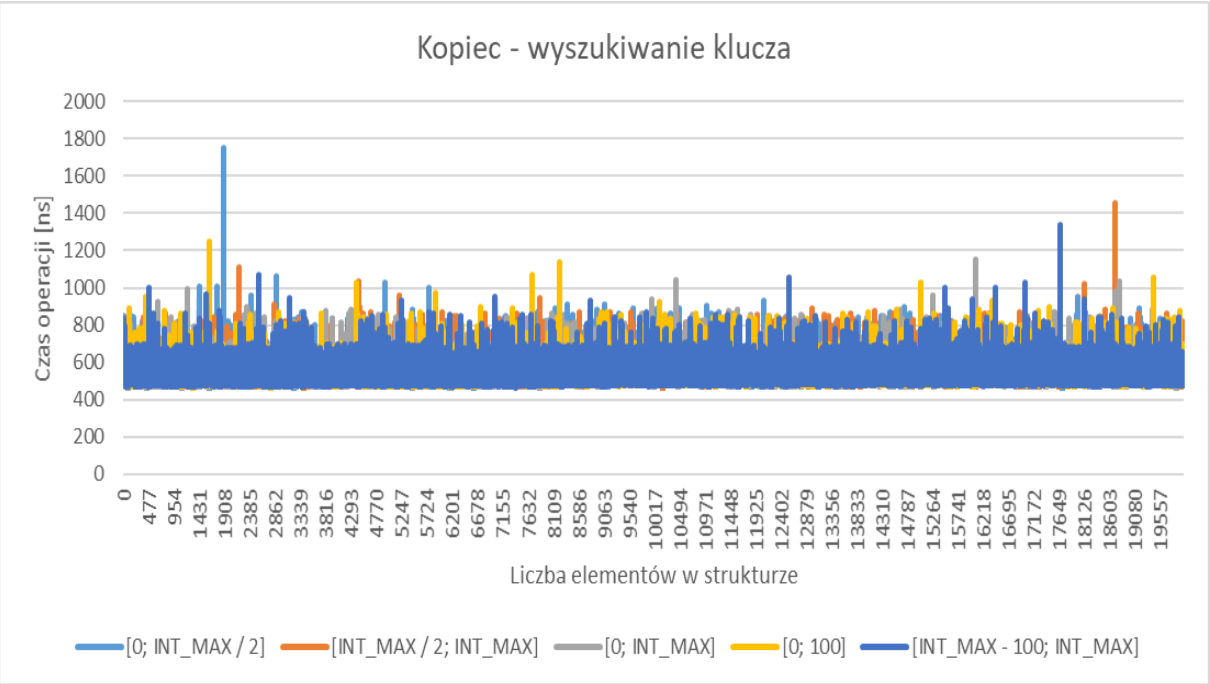
Rysunek 32 Linie trendu dla operacji usuwania klucza z kopca

Otrzymane wyniki różnią się od założeń teoretycznych. Czas dodawania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek powolnego tempa wzrostu czasu wykonywania operacji o złożoności logarytmicznej. W przeciwieństwie do operacji dodawania można zauważyć tutaj małą tendencję wzrostową.

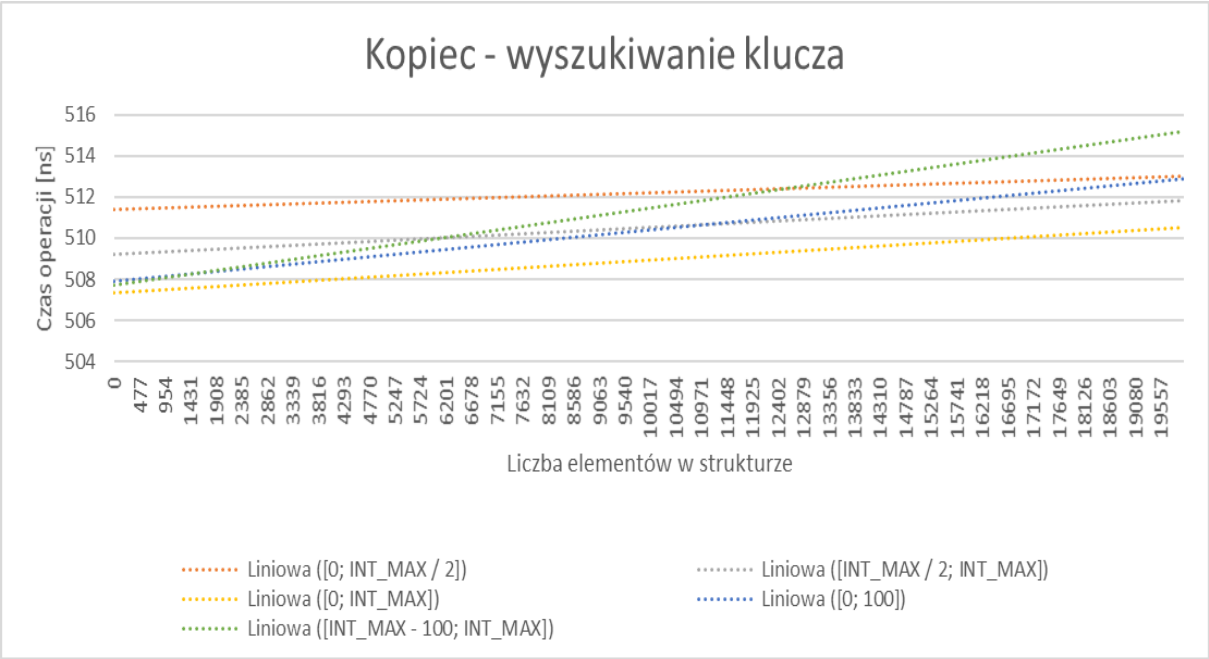
3.3.3. Wyszukiwanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	541	492	665	488	494
1000	490	492	488	483	486
2000	484	481	483	663	480
3000	482	483	492	483	489
4000	482	618	478	493	488
5000	479	481	507	645	486
6000	490	485	699	486	484
7000	491	654	487	482	489
8000	497	483	655	489	490
9000	485	677	489	486	490
10000	489	481	491	489	489
11000	616	695	486	480	494
12000	481	482	483	490	500
13000	485	485	487	487	490
14000	490	494	626	494	488
15000	482	483	486	484	489
16000	483	489	493	482	490
17000	628	496	483	482	487
18000	485	481	483	487	490
19999	485	489	493	495	500

Tabela 17 Czas [ns] wyszukiwania klucza w kopcu w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 33 Wykres zawierający wszystkie punkty pomiarowe dla operacji wyszukiwania klucza w kopcu



Rysunek 34 Linie trendu dla operacji wyszukiwania klucza w kopcu

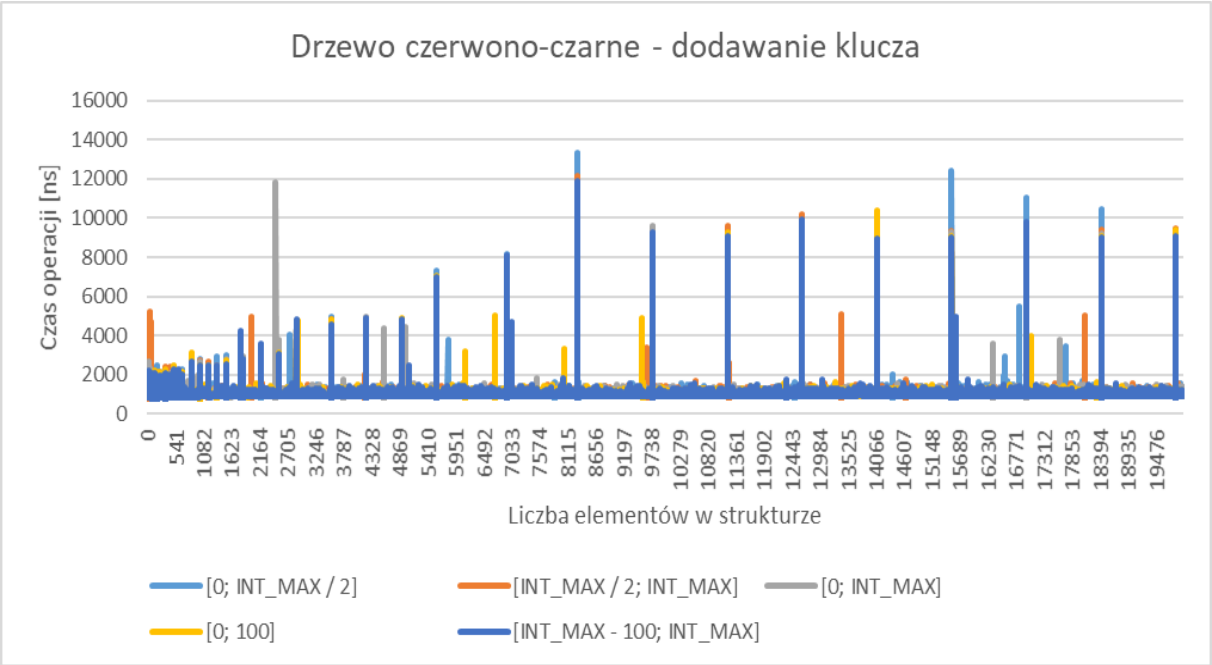
Otrzymane wyniki różnią się od założeń teoretycznych. Czas wyszukiwania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek zastosowanych rozwiązań w architekturze procesora (Intel), które pozwoliły na bardzo wydajny odczyt danych z pamięci (wyszukiwanie nie modyfikuje elementów w strukturze).

3.4. Drzewo czerwono-czarne

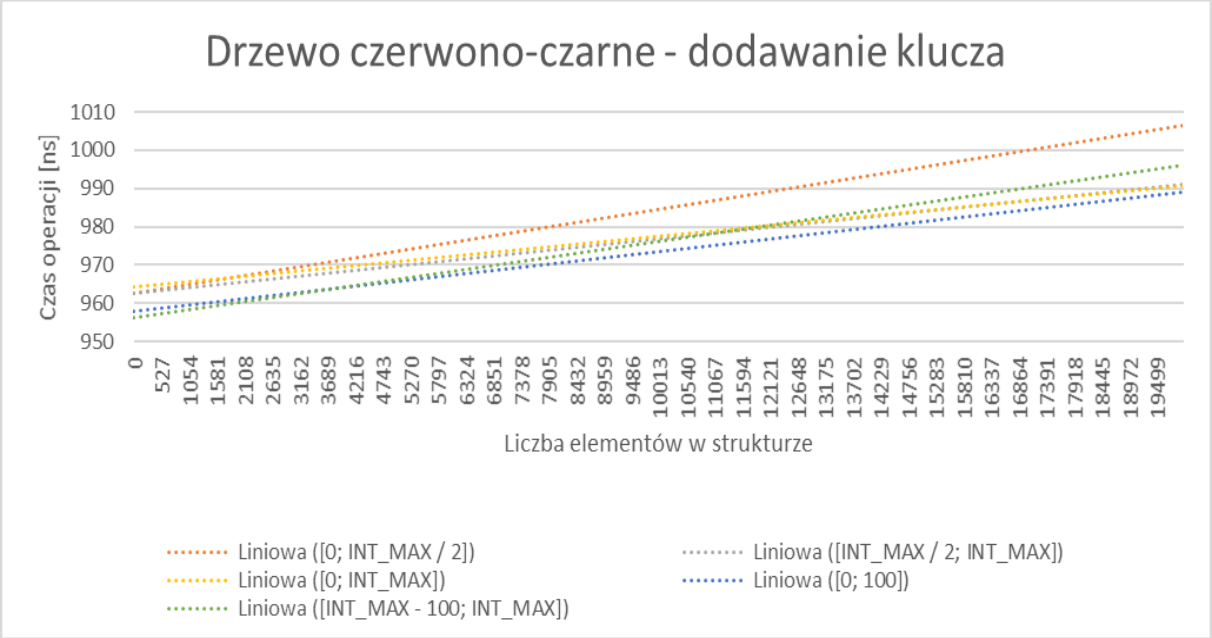
3.4.1. Dodawanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	914	1033	877	1053	1012
1000	895	1195	880	873	1207
2000	879	876	883	888	890
3000	907	902	902	893	899
4000	1104	929	911	1061	1116
5000	1057	920	930	899	1044
6000	1080	1045	907	916	1056
7000	913	903	912	899	891
8000	916	900	905	906	898
9000	901	904	901	910	898
10000	909	1412	918	909	903
11000	911	1044	914	910	910
12000	908	1095	921	918	1042
13000	916	923	915	918	912
14000	921	924	1054	1088	925
15000	927	919	907	925	919
16000	924	915	1108	909	1047
17000	930	918	911	916	920
18000	932	932	1063	912	922
19999	953	936	931	947	927

Tabela 18 Czas [ns] dodawania klucza w drzewie czerwono-czarnym w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 35 Wykres zawierający wszystkie punkty pomiarowe dla operacji dodawania klucza w drzewie czerwono-czarnym



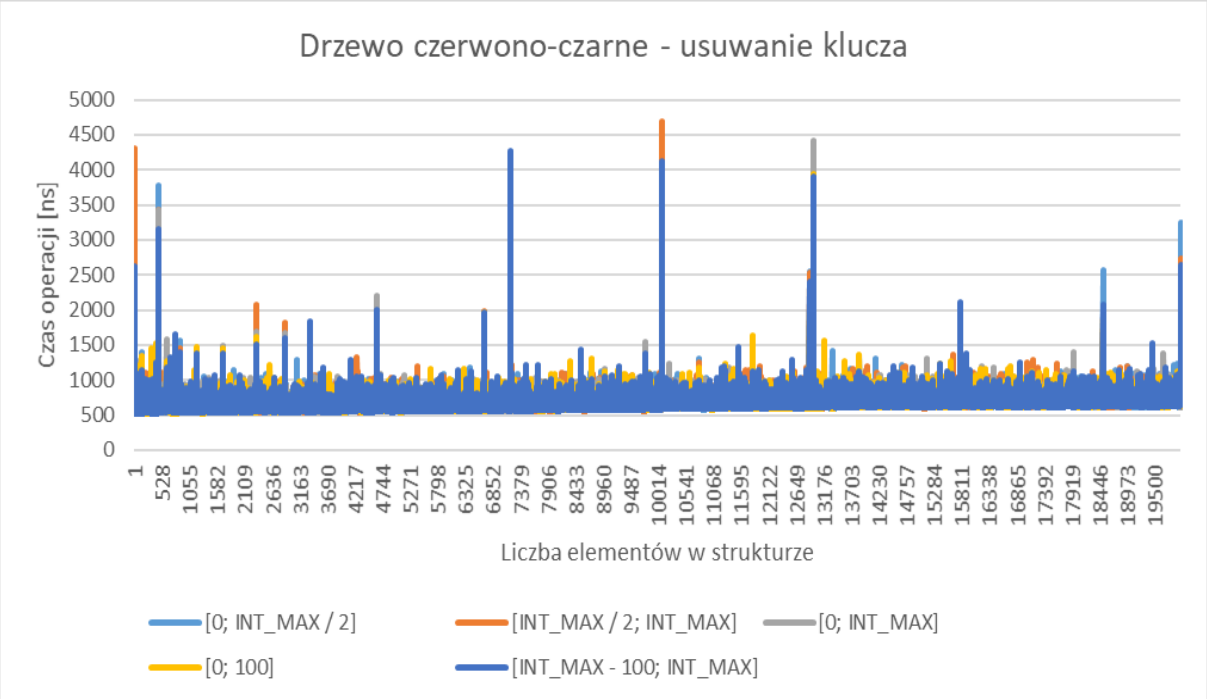
Rysunek 36 Linie trendu dla operacji dodawania klucza w drzewie czerwono-czarnym

Otrzymane wyniki różnią się od założeń teoretycznych. Czas dodawania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek powolnego tempa wzrostu czasu wykonywania operacji o złożoności logarytmicznej. Można zaobserwować jednak małą tendencję wzrostową, która może sugerować złożoność logarytmiczną operacji (jest to wolny wzrost).

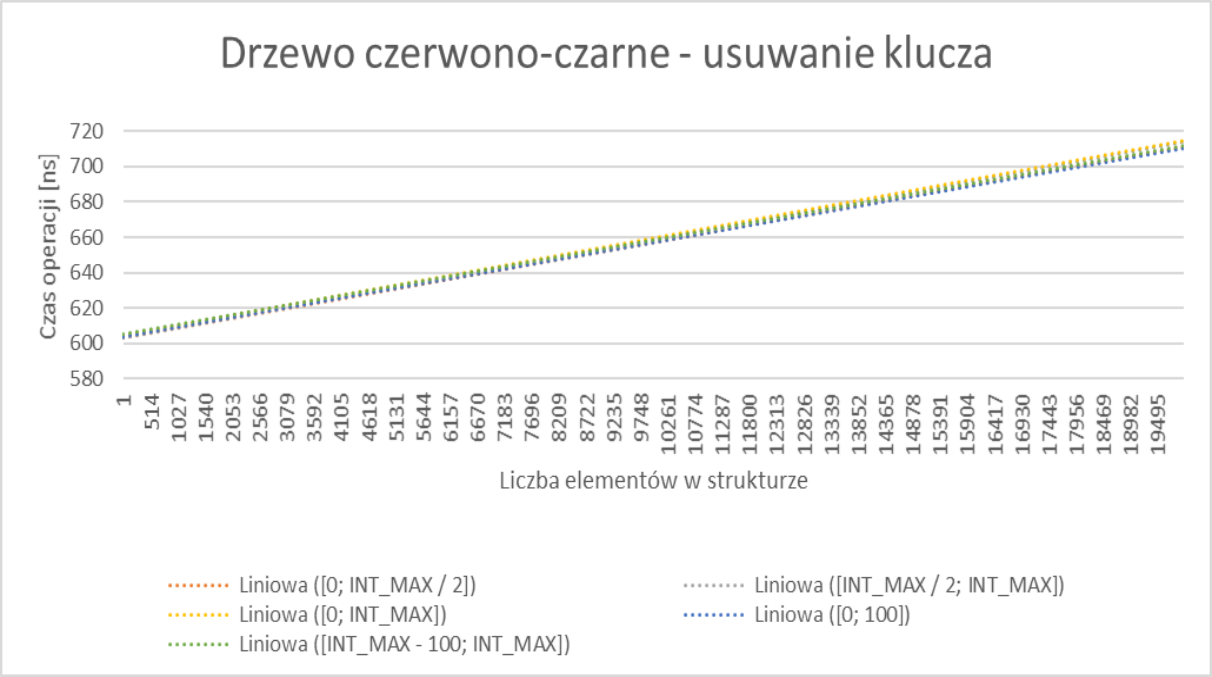
3.4.2. Usuwanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	553	553	540	533	555
1000	626	623	616	626	625
2000	639	633	631	627	782
3000	753	559	565	563	561
4000	798	844	811	787	984
5000	572	569	576	575	574
6000	586	717	591	577	584
7000	654	649	648	779	663
8000	701	709	717	700	701
9000	649	662	804	807	668
10000	683	675	671	683	1023
11000	665	672	677	808	663
12000	962	777	969	853	781
13000	676	656	667	669	671
14000	697	672	695	709	692
15000	816	855	683	678	673
16000	743	742	752	757	766
17000	703	699	693	693	696
18000	754	902	743	732	1014
19999	854	716	718	729	709

Tabela 19 Czas [ns] usuwanie klucza w drzewie czerwono-czarnym w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 37 Wykres zawierający wszystkie punkty pomiarowe dla operacji usuwanie klucza w drzewie czerwono-czarnym



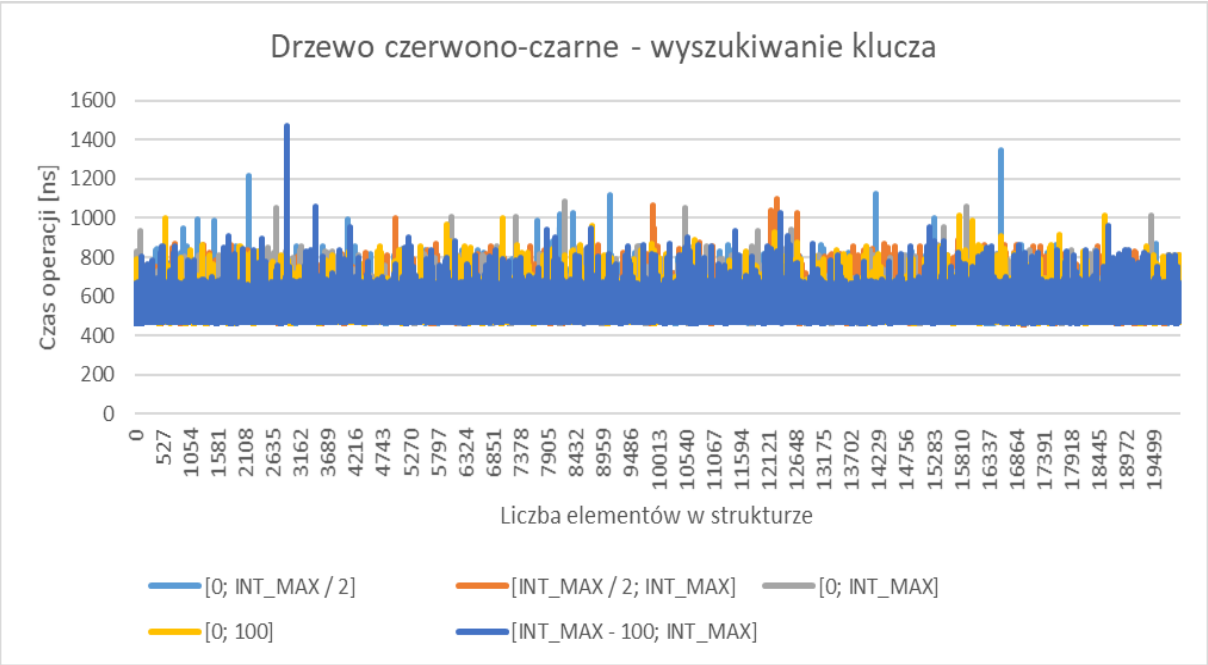
Rysunek 38 Linie trendu dla operacji usuwanie klucza w drzewie czerwono-czarnym

Otrzymane wyniki różnią się od założeń teoretycznych. Czas dodawania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek powolnego tempa wzrostu czasu wykonywania operacji o złożoności logarytmicznej. Można zaobserwować jednak tendencję wzrostową, która może sugerować złożoność logarytmiczną operacji (jest to wolny wzrost).

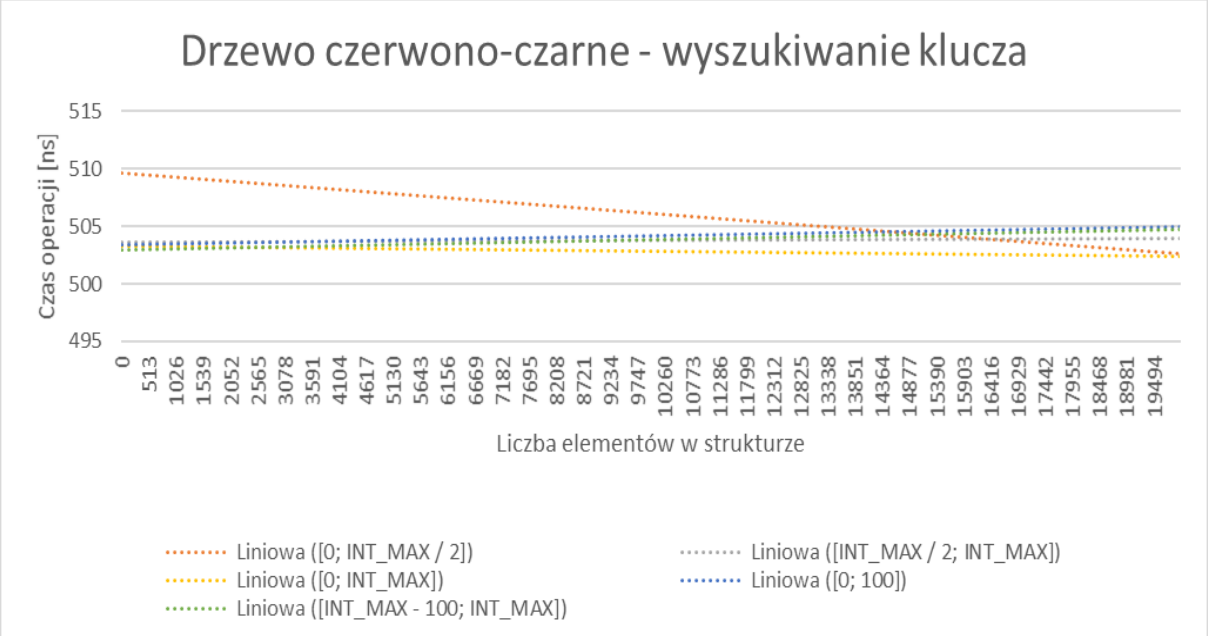
3.4.3. Wyszukiwanie klucza

Ilosc elementow w strukturze	[0; INT_MAX / 2]	[INT_MAX / 2; INT_MAX]	[0; INT_MAX]	[0; 100]	[INT_MAX - 100; INT_MAX]
1	527	487	488	490	491
1000	487	481	485	482	482
2000	477	484	476	487	477
3000	487	483	476	485	482
4000	486	489	480	487	482
5000	487	479	479	619	483
6000	480	480	482	484	482
7000	642	495	476	485	482
8000	486	479	476	485	649
9000	486	493	486	482	483
10000	473	489	468	475	473
11000	482	480	615	675	483
12000	482	483	494	487	478
13000	666	485	479	811	481
14000	481	475	473	853	489
15000	481	477	488	672	483
16000	486	490	677	500	477
17000	622	478	621	481	478
18000	485	672	487	480	620
19999	482	484	481	476	492

Tabela 20 Czas [ns] wyszukiwania klucza w drzewie czerwono-czarnym w zależności od l. elementów i wartości kluczy (równomierny wybór 20 punktów pomiarowych)



Rysunek 40 Wykres zawierający wszystkie punkty pomiarowe dla operacji wyszukiwania klucza w drzewie czerwono-czarnym



Rysunek 39 Linie trendu dla operacji wyszukiwania klucza w drzewie czerwono-czarnym

Otrzymane wyniki różnią się od założeń teoretycznych. Czas wyszukiwania elementu w tej strukturze okazał być się stały (O(1)) - w granicach niepewności pomiarowych. Pomiary powtórzono dla 1.000.000 elementów, jednak uzyskane wyniki nadal wskazywały na stały czas operacji. Może być to skutek zastosowanych rozwiązań w architekturze procesora (Intel), które pozwoliły na bardzo wydajny odczyt danych z pamięci (wyszukiwanie nie modyfikuje elementów w strukturze).

4. Podsumowanie

Badane struktury można podzielić na dwie grupy:

- działające w czasie liniowym w pesymistycznych przypadkach (tablica, lista)
- działające w czasie logarytmicznym w pesymistycznych przypadkach (kopiec, drzewo czerwono – czarne)

Lista okazała się górować nad tablicą, gdyż w operacjach dodawania na oraz usuwania z początku uzyskała czas O(1), gdzie tablica potrzebuje czasu O(n) na wykonanie ww. operacji. Dodanie/usunięcie elementu w losowym miejscu wymaga liniowego czasu w obu strukturach– jest to spowodowane koniecznością wyszukania elementu w pierwszej kolejności (lista) lub przesunięcia danych (tablica). Dodanie/usuniecie elementu z końca w obu strukturach okazało być się operacją stałą. Wyszukiwanie wymaga teoretycznego czasu liniowego zarówno w tablicy jaki i liście, jednak wykazano, że w pewnych okolicznościach (szukanie losowego elementu z zakresu, przy odpowiednio małej wielkości struktury oraz na specyficznym sprzęcie) może być operacją w praktyce (średnio) stałą (chociaż jest to sprzeczne z teorią). Lista wydaje się być odpowiednim kandydatem na implementację kolejki, zaś tablica sprawdziłaby się w implementacji stosu. Warto nadmienić, że przewagą tablicy nad listą jest natychmiastowy dostęp do elementu o znanym indeksie, który to w wielu przypadkach decyduje o słuszności zastosowania właśnie tablicy, a nie listy do przechowywania danych.

Kopiec i drzewo czerwono – czarne potrzebują w teorii logarytmicznego czasu na wykonanie wszystkich badanych operacji słownikowych (z wyjątkiem operacji wyszukiwania w kopcu – ta ma złożoność O(n)). Badania wykazały jednak, że przy odpowiednich rozmiarach danych i konfiguracji sprzętu operacje takie, jak dodawanie, usuwanie czy wyszukiwanie elementu można traktować jako działające (średnio) w stałym czasie. Wynika to z faktu, iż przyrost czasu potrzebnego no wykonanie ww. operacji dla coraz większej ilości danych jest wolny. Niemniej jednak należy podkreślić, że drzewo czerwono – czarne ma zasadniczą przewagę nad kopcem: jest to zrównoważone drzewo poszukiwań binarnych gwarantujące górne ograniczenie na operację wyszukiwania elementu równe O(lgn). Kosztem tego stanu rzeczy jest spory czynnik stały w operacjach dodawania i usuwania drzewa, co uwidacznia się na wykresach w postaci wyższej dolnej granicy przedziału wartości wymaganego na wykonanie operacji czasu. Kopiec wydaje się być dobrym kandydatem na bazę dla kolejki priorytetowej ze względu na swoją własność kopca. Drzewo czerwono – czarne pozwoli natomiast na szybkie wyszukiwanie elementu; jego implementacja ma jednak sens, gdy ilość danych jest względnie duża.

Warto zauważyć, iż na wykresach wystąpiły nagłe skoki wartości. Ich geneza może być różna. W tablicy i kopcu wynikają one z czasu potrzebnego na realokację danych przy zwiększaniu rozmiaru, chociaż mogą być to również opóźnienia wynikłe z działania systemu operacyjnego, który podczas wykonywania testów nie zawsze przydzielał najwyższy priorytet zadaniom programu testującego. W przypadku listy i drzewa czerwono – czarnego nieciągłości tych jest więcej w porównaniu z tablicą i kopcem. Może to wynikać z faktu, iż do operacji na tych strukturach używa się w dużej mierze wskaźników, przez co pojawia się wiele żądań pośredniego odczytu z pamięci podczas działania programu, co może skutkować zwiększonym wykorzystaniem magistral adresowych i w ostateczności w zaobserwowany sposób wpłynąć na czas wykonywania operacji.

Charakterystyczne są również duże zagęszczenia w dolnych przedziałach czasowych operacji wyszukiwania we wszystkich badanych strukturach. Jest to następstwem wyszukiwania kluczy z rozkładu równomiernego i uwidacznia się to na wykresach w postaci „równomiernych” czasów potrzebnych na znalezienie danego klucza w strukturze.