



Politechnika Wrocławska

**Badanie efektywności operacji na
danych w podstawowych
strukturach danych**

Łukasz Wdowiak

Prowadzący: dr inż. Dariusz Banasiak

Grupa: K03-37h, Pon 15:15-16:55 TP

Wydział Informatyki i Telekomunikacji

Informatyka Techniczna

IV semestr

1 Wstęp Teoretyczny

Złożoność obliczeniowa jest jednym z podstawowych pojęć w informatyce. Opisuje ona ilość zasobu czasu, czy pamięci komputera potrzebnych do wykonania pewnej operacji (algorytmu). Złożoność obliczeniową wyraża się w notacji dużego O (notacja asymptotyczna), która mówi nam o tym jak szybko rośnie ilość zasobów potrzebna do wykonania algorytmu w najgorszym przypadku. Dobra znajomość struktur danych jest przydatną umiejętnością w dziedzinie informatyki. Programując swoje programy powinniśmy umieć dopasować do dla niego jak najlepszą strukturę danych, aby program mógł działać w optymalny sposób. Tablica dynamiczna jest strukturą, która zajmują ciągły obszar w pamięci i może zmieniać swój rozmiar, w przeciwieństwie do tablicy statycznej. Lista dwukierunkowa nie jest ciągła w pamięci komputera. Jest ona zbudowana z tak zwanych węzłów, które przechowują jego wartość oraz wskaźnik na poprzedni i następny element listy. Z kolei kopiec jest już strukturą reprezentującą drzewo binarne, w której wszystkie poziomy muszą być wypełnione z wyjątkiem ostatniego. Wyróżniamy dwa rodzaje kopców: maksymalny i minimalny. W kopcu maksymalnym w korzeniu przechowujemy węzeł o największej wartości i analogicznie jest w kopcu minimalnym, gdzie w korzeniu przechowujemy węzeł o najmniejszej wartości. Drzewo czerwono-czarne jest przykładem drzewem samobalansującego się, gdzie węzły są kolorowane odpowiednio kolorem czarnym lub czerwonym. Drzewo takie posiada kilka własności: korzeń drzewa zawsze jest czarny, każdy liść jest czarny, każdy czerwony węzeł ma czarne tylko dzieci, dla każdego węzła wszystkie ścieżki do liścia zawierają taką samą liczbę czarnych węzłów.

1.1 Tablica dynamiczna

funkcja	złożoność obliczeniowa
dodanie elementu	$O(n)$
usunięcie elementu	$O(n)$
wyszukanie elementu	$O(n)$

Tabela 1: złożoność obliczeniowa tablicy dynamicznej

1.2 Lista dwukierunkowa

funkcja	złożoność obliczeniowa
dodanie elementu	$O(n)$
usunięcie elementu	$O(n)$
dodanie elementu na początku	$O(1)$
dodanie elementu na końcu	$O(1)$
wyszukanie elementu	$O(n)$

Tabela 2: złożoność obliczeniowa listy dwukierunkowej

1.3 Kopiec maksymalny

funkcja	złożoność obliczeniowa
dodanie elementu	$O(\log(n))$
usunięcie elementu	$O(\log(n))$
wyszukanie elementu	$O(n)$

Tabela 3: złożoność obliczeniowa kopca maksymalnego

1.4 Drzewo czerwono-czarne

funkcja	złożoność obliczeniowa
dodanie elementu	$O(\log(n))$
usunięcie elementu	$O(\log(n))$
wyszukanie elementu	$O(n)$

Tabela 4: złożoność obliczeniowa drzewa czerwono-czarnego

2 Plan projektu

Projekt implementuje cztery podstawowe struktury danych takich jak:

- Tablica dynamiczna
- Lista dwukierunkowa
- Kopiec maksymalny
- Drzewo czerwono-czarne

2.1 Założenia dotyczące struktur danych:

- Struktury przechowują 4 bajtową zmienną ze znakiem (typ int)
- Wszystkie struktury zajmują możliwie najmniejszą ilość miejsca w pamięci
- Do tablicy dynamicznej i listy dwukierunkowej zaimplementowane zostały dodatkowo metody odpowiedzialne za dodawanie/usuwanie na początku i końcu struktury, jak i metoda odpowiedzialna za dodanie elementu w dowolne miejsce

2.2 Założenia dotyczące programu:

- Program został napisany obiektowo w języku C++
- Z pozycji menu jesteśmy w stanie osobno przetestować działanie każdej ze struktur
- Generator liczb pseudolosowych oparty jest o generator Mersenne Twister 19937, który został zaimportowany z biblioteki standardowej
- Liczby generowane przez generator nie powtarzają się
- Czas mierzony jest za pomocą funkcji ze standardowej biblioteki

```
std::chrono::high_resolution_clock::now();
```

- Do przeprowadzenia eksperymentów z pomiarem czasu wykonania operacji zaimplementowano dodatkowy tryb, który automatycznie testuje nam struktury i zapisuje wyniki pomiarów czasu do pliku.
- Czas przedstawiony będzie w nanosekundach [ns]

2.3 Założenia dotyczące testowania struktur:

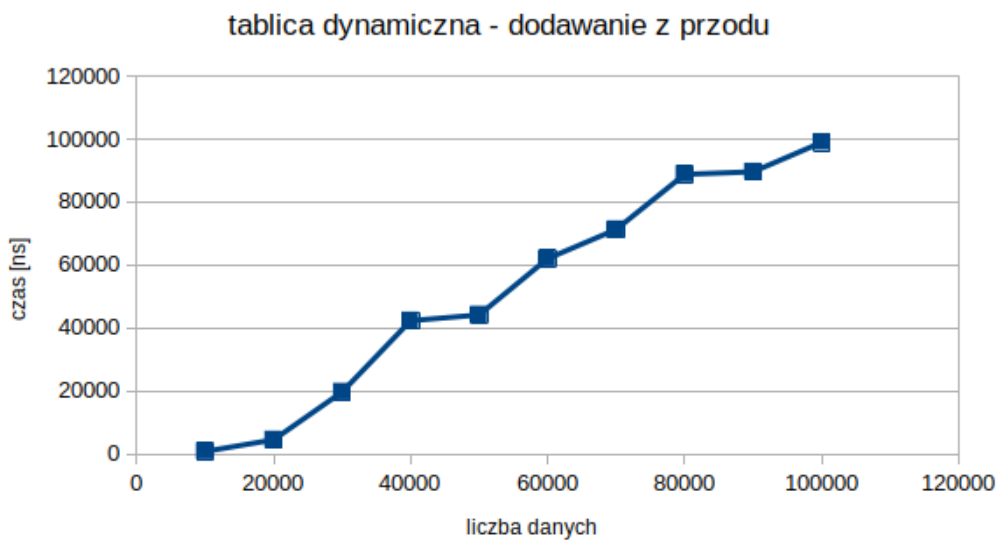
- Testowane rozmiary struktur od 10000 do 100000 co 10000 elementów
- Do analizy eksperymentów została wzięta średnia arytmetyczna 100 pomiarów każdego rozmiaru (za każdym razem generowany został nowy zestaw danych)

3 Wyniki eksperymentów

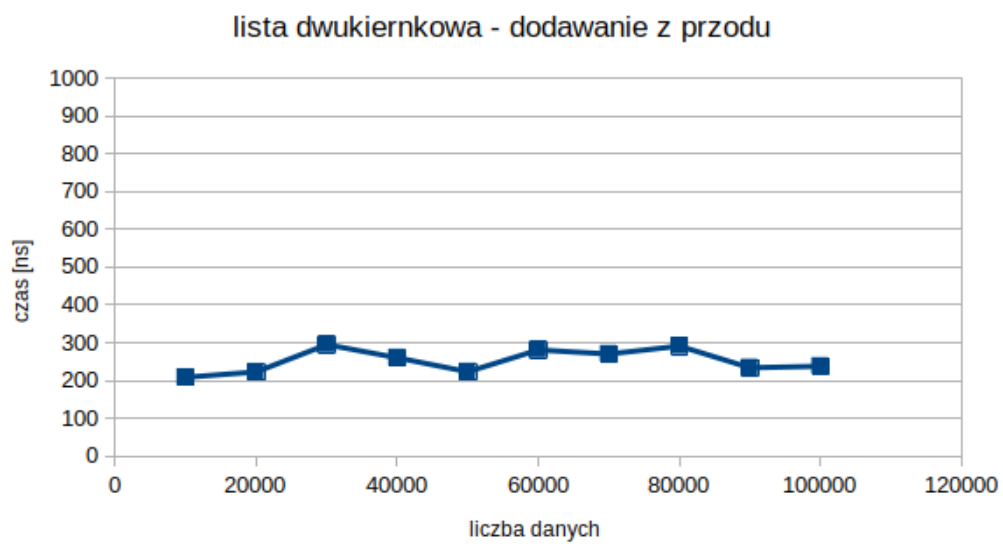
3.1 Operacja dodania elementu

liczba danych	tablica dynamiczna	lista dwukierunkowa	kopiec maksymalny	drzewo czerwono-czarne
10000	894.12	208.77	1009.71	313.58
20000	4582.97	222.93	4853.98	527.11
30000	19762.1	295.12	13691.4	959.24
40000	42390	260.3	24676.8	2165.46
50000	44213.3	223.25	30141.8	2234.73
60000	62082.7	281.06	31173.5	2332.85
70000	71345.2	270.37	45089.7	2438.29
80000	88825.1	291.22	46598.5	2367.59
90000	89643	233.9	55011.8	2415.49
100000	98981	237.98	58527.7	2453.21

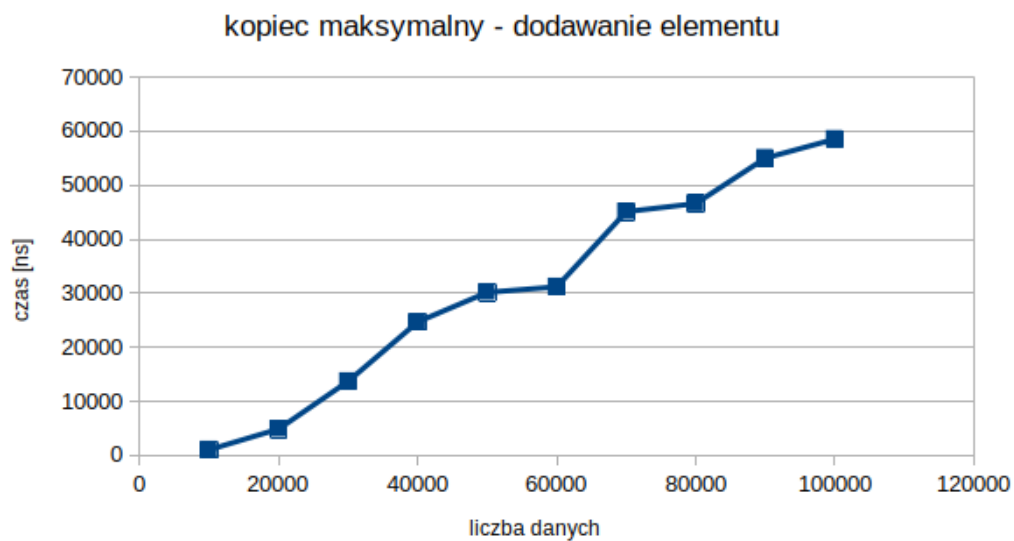
Tabela 5: czasy operacji dodania elementów



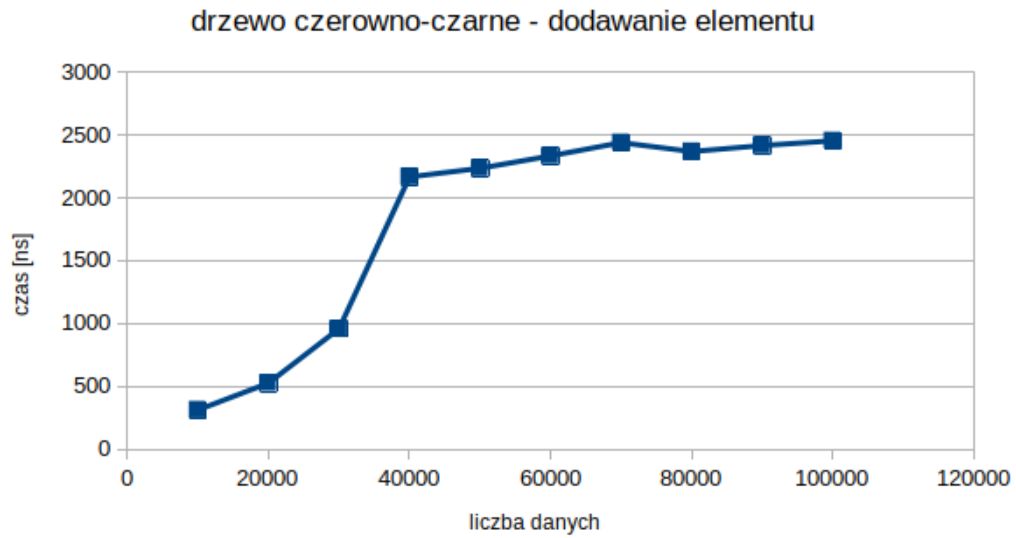
Rysunek 1: tablica dynamiczna - dodawanie elementu



Rysunek 2: lista dwukierunkowa - dodawanie elementu



Rysunek 3: kopiec maksymalny - dodawanie elementu

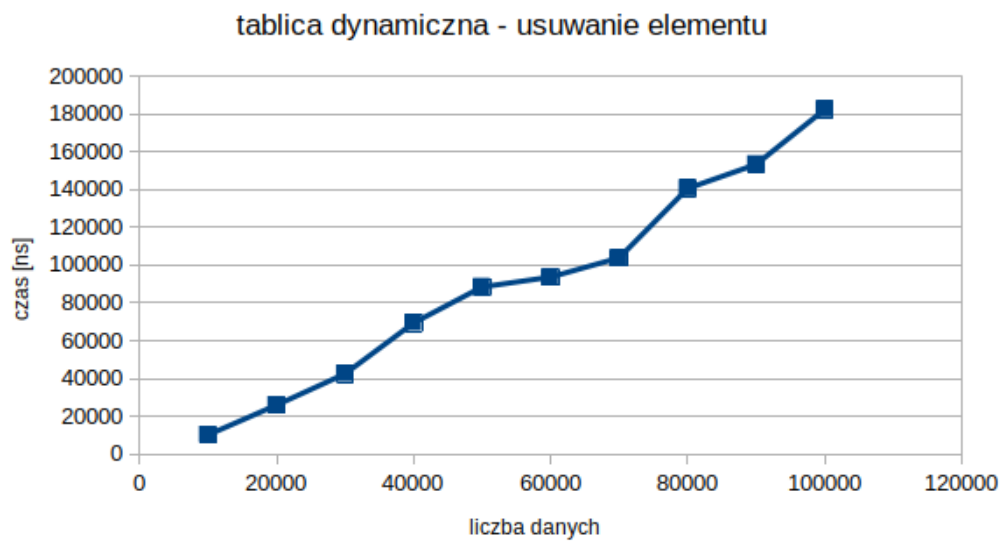


Rysunek 4: drzewo czerowno-czarne - dodawanie elementu

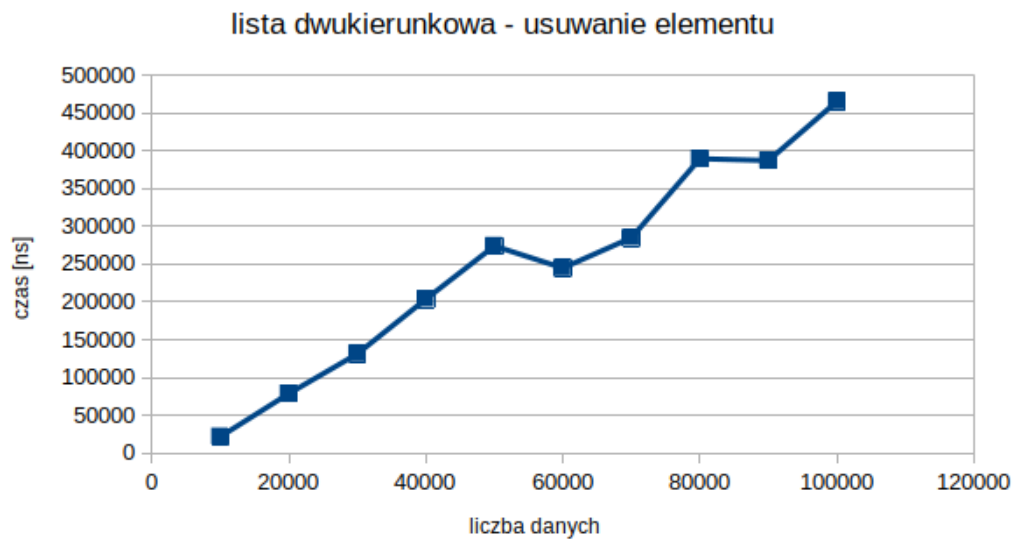
3.2 Operacja usunięcia elementu

liczba danych	tablica dynamiczna	lista dwukierunkowa	kopiec maksymalny	drzewo czerwono-czarne
10000	10225.2	21945	1297.55	732.9
20000	26015.5	78834.4	3931.37	1056.74
30000	42553.9	131368	8619.38	1501.48
40000	69443.3	203799	17839	1690.59
50000	88490.7	274096	24641.1	2049.07
60000	93686.7	244940	29241	2055.55
70000	103887	284894	30115.8	2273.32
80000	140575	389400	43803.3	2332.38
90000	153316	387107	50502.8	2264.43
100000	182353	465196	55283.7	2301.55

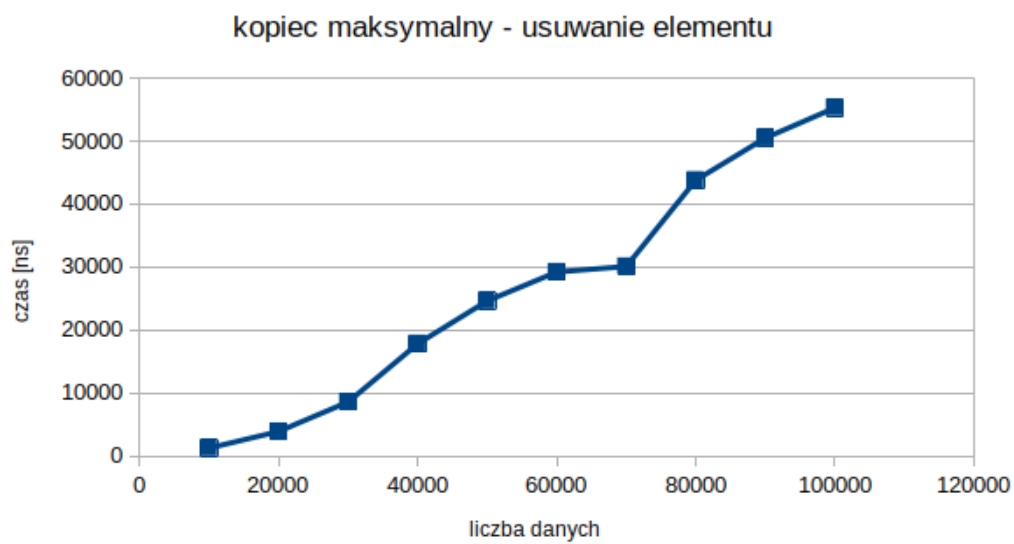
Tabela 6: czasy operacji usunięcia elementów



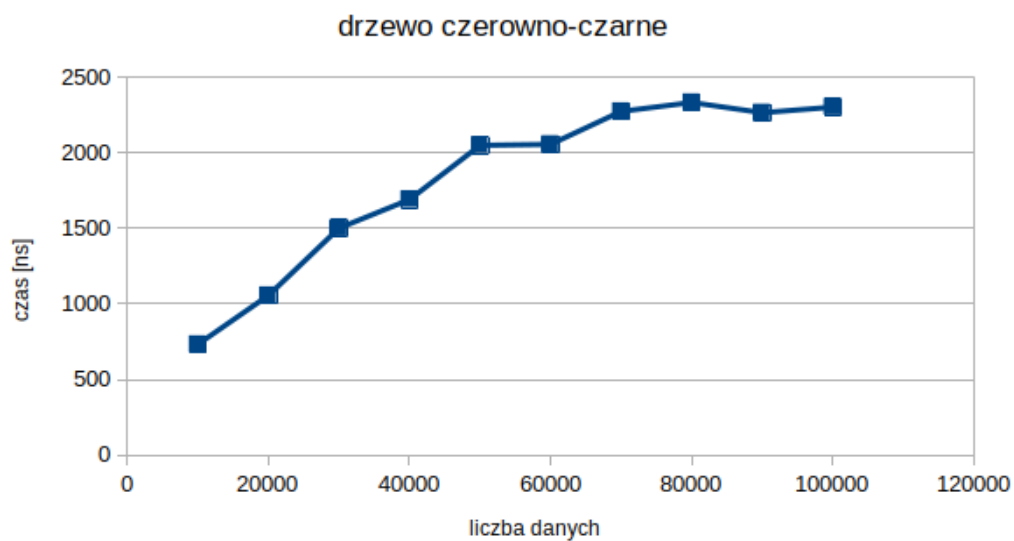
Rysunek 5: tablica dynamiczna - usuwanie elementu



Rysunek 6: lista dwukierunkowa - usuwanie elementu



Rysunek 7: kopiec maksymalny - usuwanie elementu

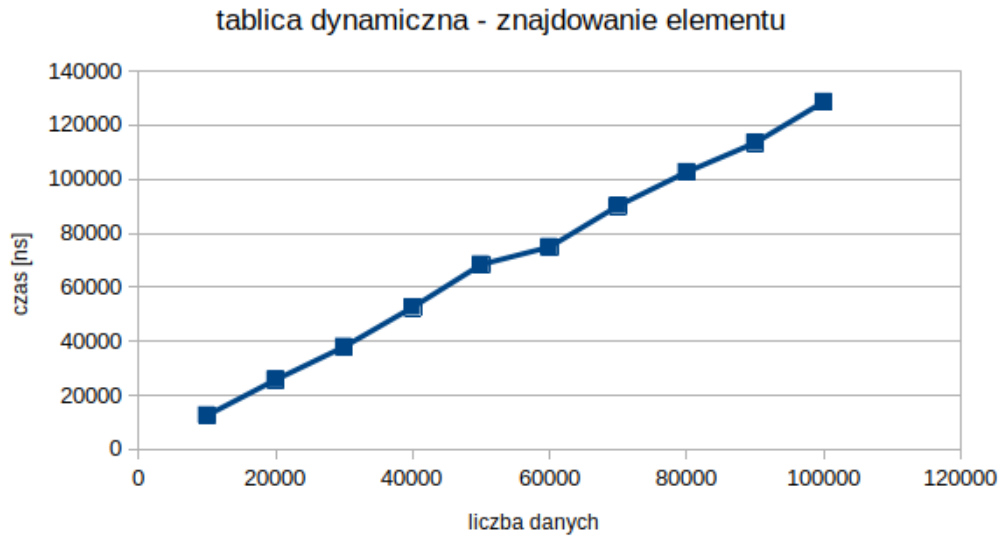


Rysunek 8: drzewo czerwono-czarne - usuwanie elementu

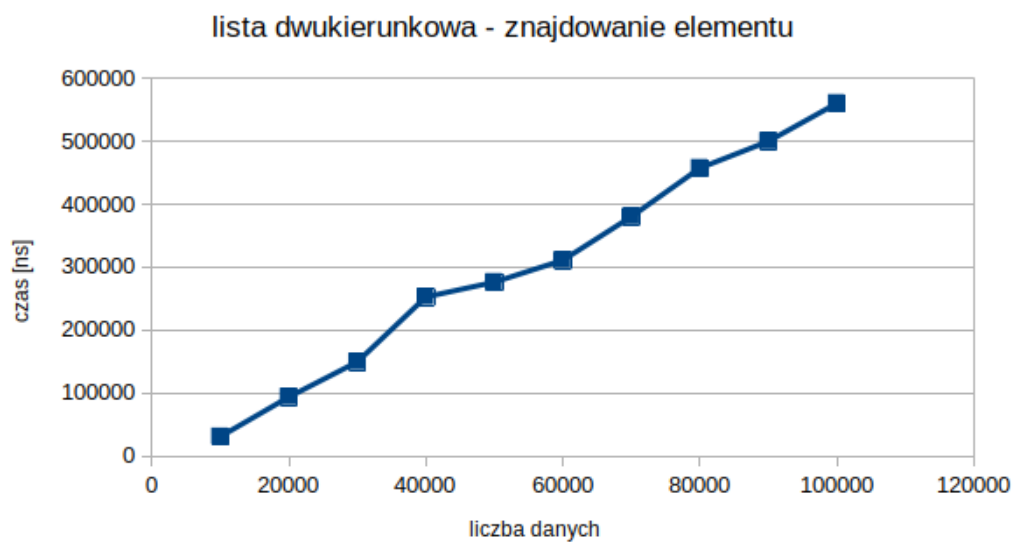
3.3 Operacja znalezienia elementu

liczba danych	tablica dynamiczna	lista dwukierunkowa	kopiec maksymalny	drzewo czerwono-czarne
10000	12535.4	31151	11183.2	678.88
20000	25693.3	93959.1	25202.3	906.62
30000	37925.2	149680	37596.7	994.45
40000	52519.6	252685	61150	1469.21
50000	68399	276284	66482.4	1540.46
60000	74876.4	311062	74439.1	1592.87
70000	90139.8	380265	77793.7	1588.4
80000	102625	457281	97497.6	1775.18
90000	113450	499711	111294	1690.2
100000	128633	560054	135791	1846.81

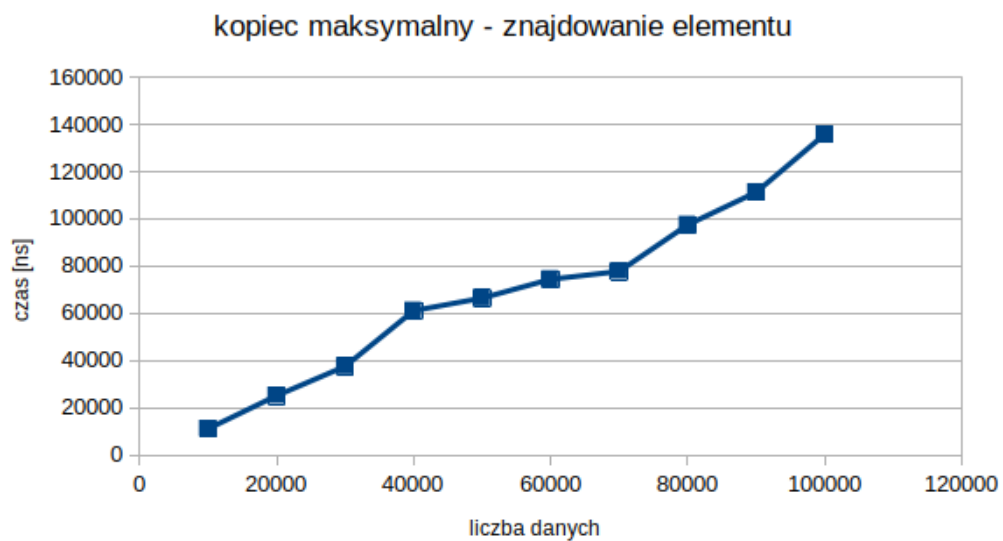
Tabela 7: czasy operacji znalezienia elementów



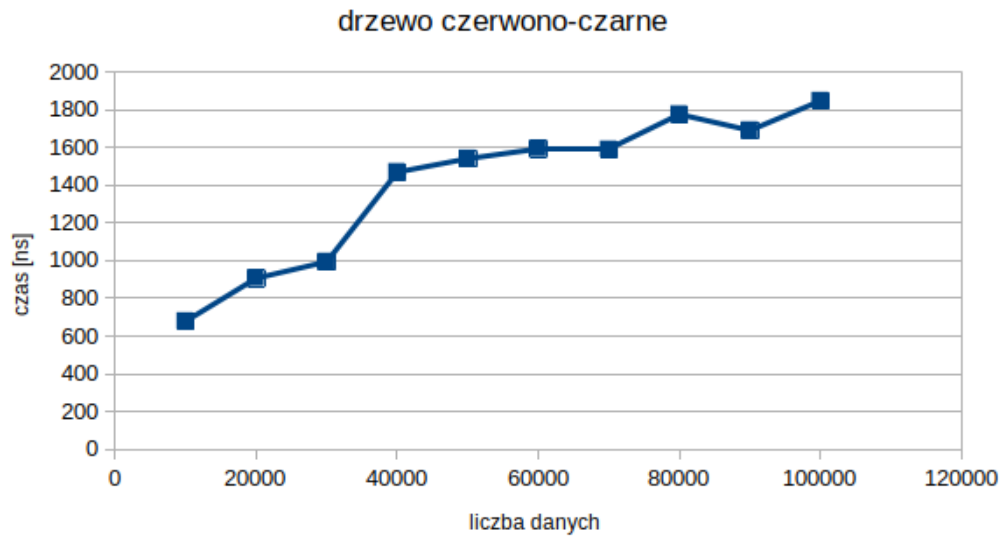
Rysunek 9: tablica dynamiczna - znajdowanie elementu



Rysunek 10: lista dwukierunkowa - znajdowanie elementu



Rysunek 11: kopiec maksymalny - znajdowanie elementu

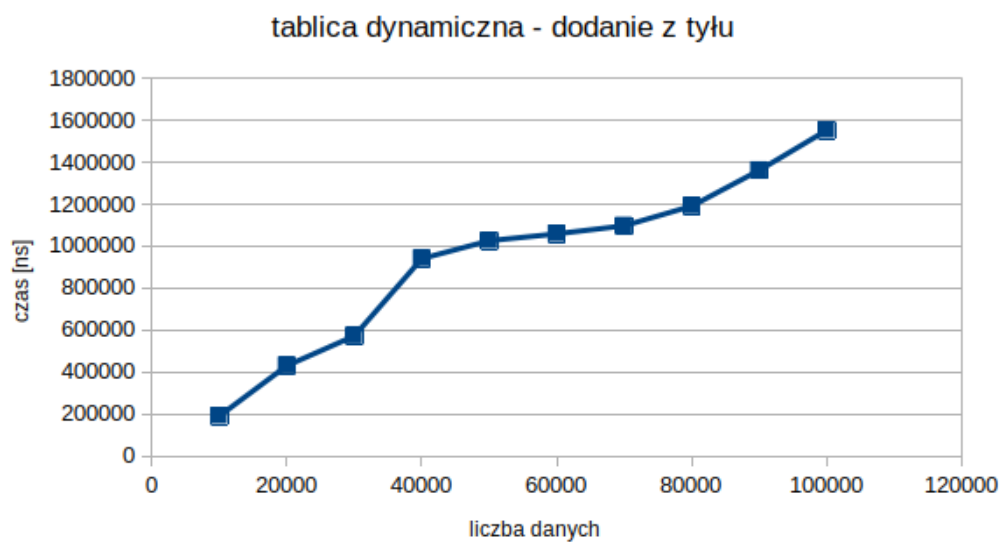


Rysunek 12: drzewo czerwono-czarne - znajdowanie elementu

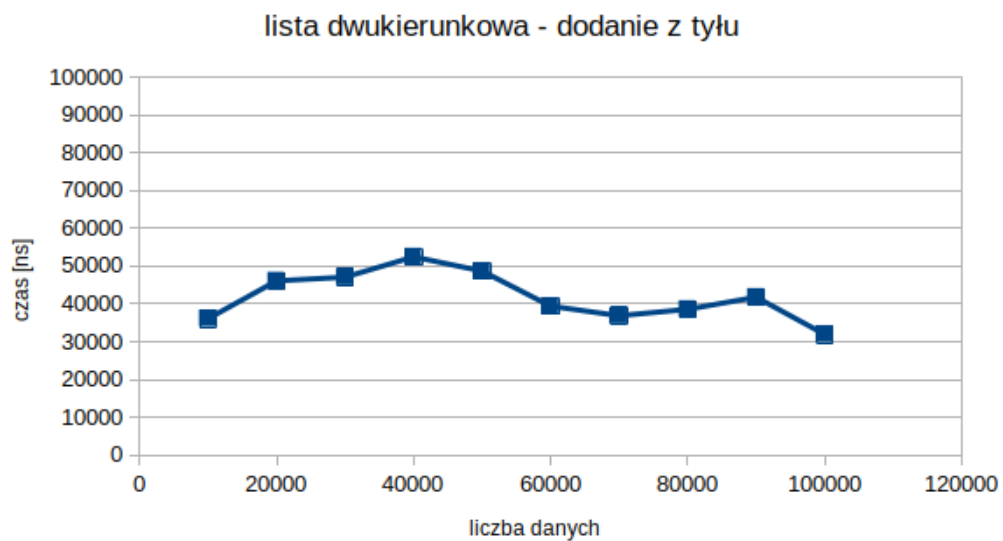
3.4 Operacja dodania elementu na koniec tablicy/listy

liczba danych	tablica dynamiczna	lista dwukierunkowa
10000	191221	36100
20000	430766	46108
30000	572395	47101
40000	940646	52475
50000	1026060	48721
60000	1058668	39420
70000	1097097	36892
80000	1191130	38591
90000	1360900	41796
100000	1551640	31956

Tabela 8: czasy operacji dodania elementów na koniec tablicy/listy



Rysunek 13: tablica dynamiczna - dodanie elementu na koniec

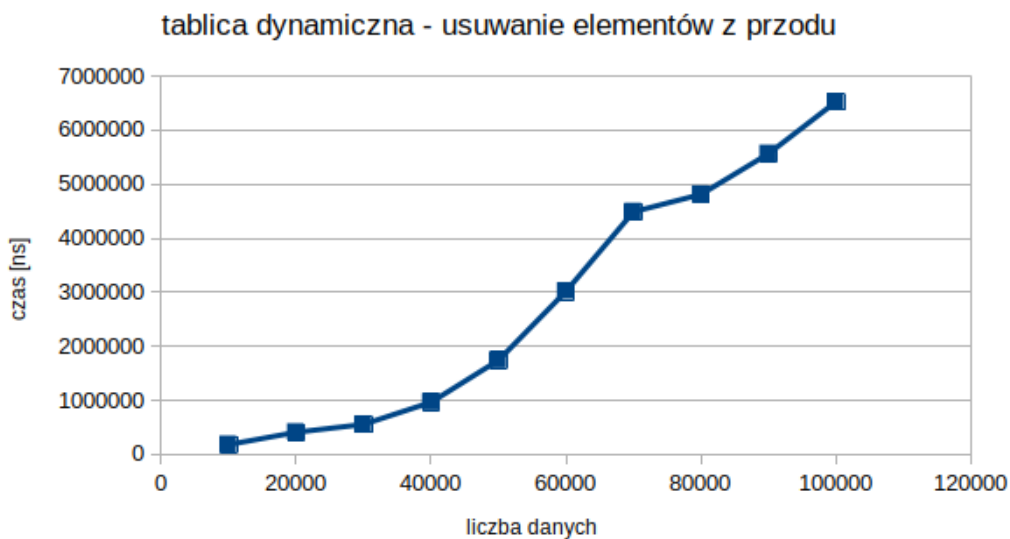


Rysunek 14: lista dwukierunkowa - dodanie elementu na koniec

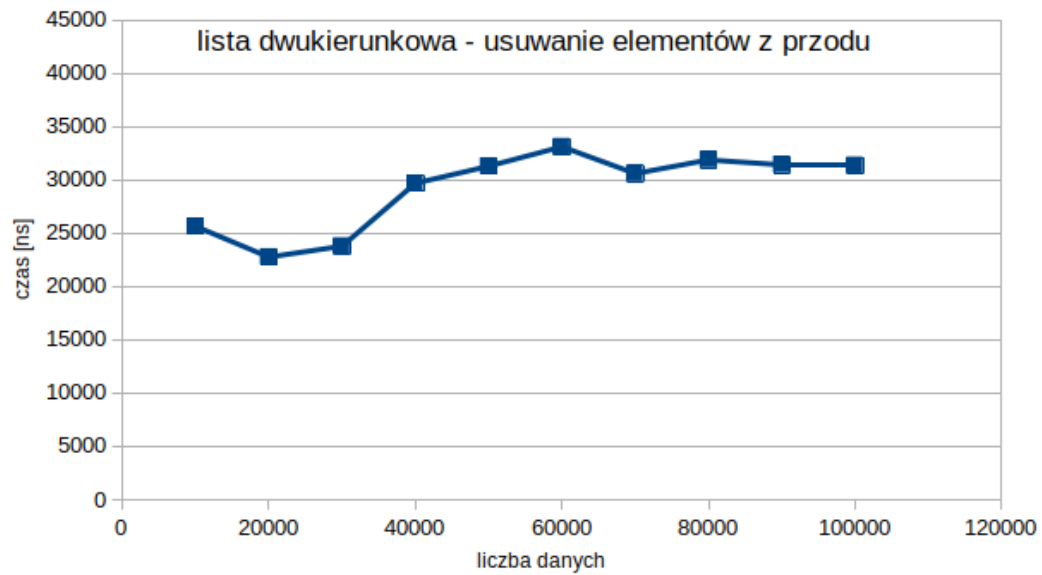
3.5 Operacja usunięcia z początku tablicy/listy

liczba danych	tablica dynamiczna	lista dwukierunkowa
10000	178443	25649
20000	405506	22764
30000	553107	23780
40000	961677	29682
50000	1740520	31280
60000	3016000	33106
70000	4488180	30590
80000	4810430	31872
90000	5563710	31407
100000	6522640	31351

Tabela 9: czasy operacji usunięcia elementów z początku tablicy/listy



Rysunek 15: tablica dynamiczna - usunięcie elementu z początku

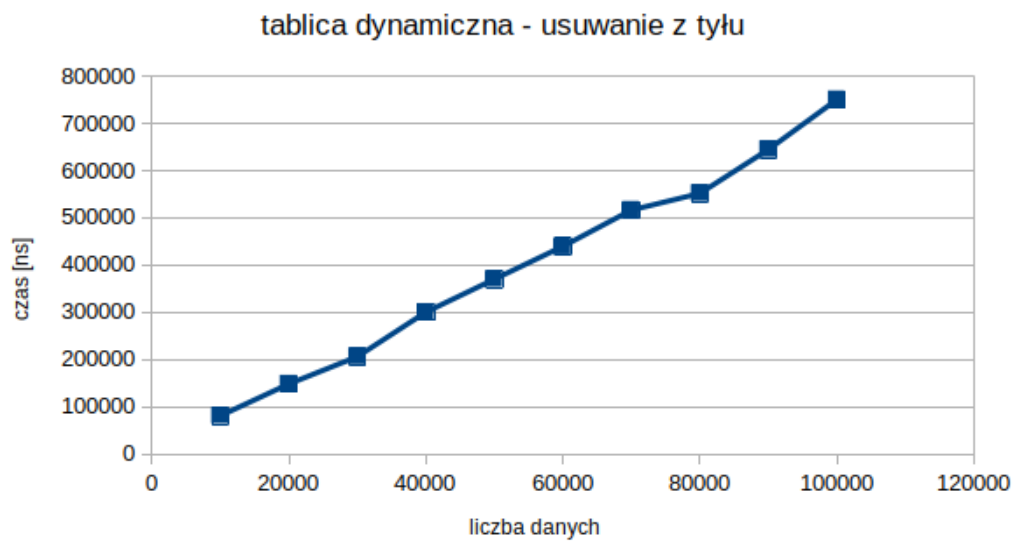


Rysunek 16: lista dwukierunkowa - usunięcie elementu z początku

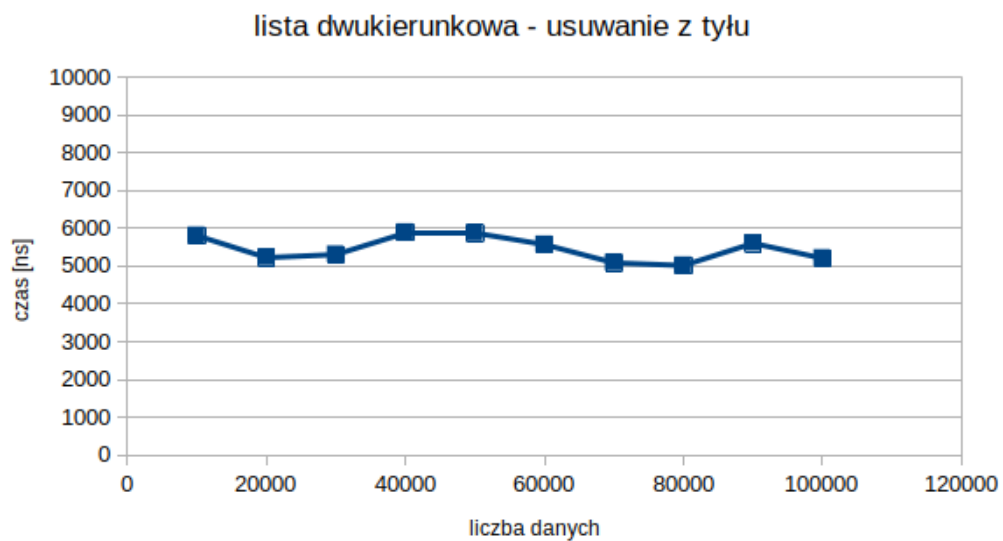
3.6 Operacja usunięcia z końca tablicy/listy

liczba danych	tablica dynamiczna	lista dwukierunkowa
10000	81364	5809
20000	148629	5226
30000	206444	5308
40000	300550	5896
50000	370301	5871
60000	439825	5572
70000	516625	5089
80000	551832	5018
90000	644766	5606
100000	750343	5209

Tabela 10: czasy operacji usunięcia elementów z końca tablicy/listy



Rysunek 17: tablica dynamiczna - usunięcie elementu z końca

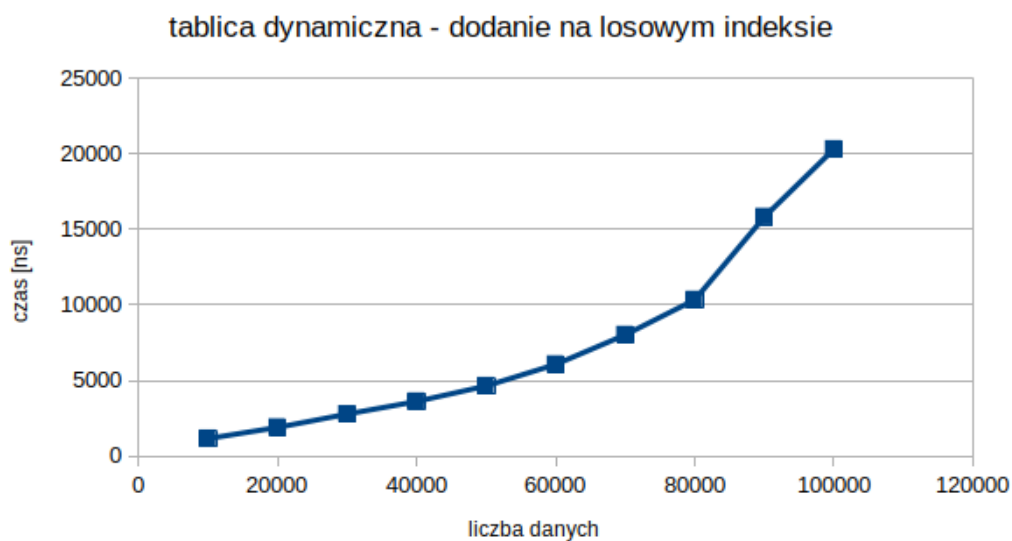


Rysunek 18: lista dwukierunkowa - usunięcie elementu z końca

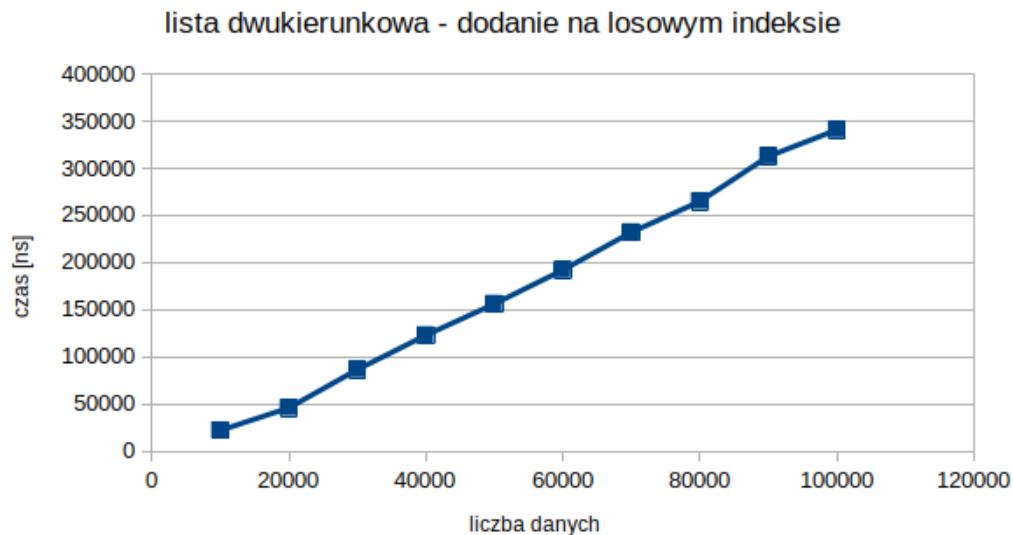
3.7 Operacja dodania elementu na losowym indeksie tablicy/listy

liczba danych	tablica dynamiczna	lista dwukierunkowa
10000	1170.27	22258.6
20000	1902.87	46134.2
30000	2788.7	86738.2
40000	3622.19	123101
50000	4646.94	156296
60000	6070.95	192237
70000	8026.6	232296
80000	10341.5	264949
90000	15825	312702
100000	20302.6	340774

Tabela 11: czasy operacji dodania elementów na losowej pozycji tablicy/listy



Rysunek 19: tablica dynamiczna - dodanie elementu na losowej pozycji



Rysunek 20: lista dwukierunkowa - dodanie elementu na losowej pozycji

4 Wnioski

Złożoność obliczeniowa różnych operacji na strukturach danych jest zgodna z przewidywanymi złożonościami pokazanymi w tabelach z jednym wyjątkiem. Wyjątkiem jest tutaj struktura zwana kopcem, gdyż do jej zaimplementowania wykorzystano ciągłą strukturę w pamięci (implementacja na tablicy). W związku z założeniem, że wszystkie struktury mają zajmować możliwe jak najmniej miejsca, to złożoność operacji dodawania czy usuwania nie przyjmują $O(\log n)$, lecz $O(n)$.

Czasy iterowania przez tablice dynamiczne są zdecydowanie mniejsze niż w przypadku iteracji przez listę dwukierunkową.

Lista dwukierunkowa łni podczas dodawania elementów na końcu czy początku tablicy z racji złożoności $O(1)$.

Zdecydowanie najefektywniejszą strukturą, którą wykazały eksperymenty jest drzewo czerwono-czarne. Czasy wykonywania operacji na tej strukturze były najmniejsze ze wszystkich testowanych struktur.

Literatura

- [1] Thomas H. Cormen, Charles E. Leiserson, Ron Rivest, Clifford Stein (2022) *Introduction to algorithms 4th edition*, MIT Press and McGraw-Hill.