**Teza**

Nasza teza jest tak że najszybszą strukturą w dodaniu jak i usuwaniu będą obie listy a wy wyszukiwaniu będzie lista dwukierunkowa, gdyż można przeszukać listę w obu kierunkach

**Plan eksperymentu**

Sprawdzenie który z implementowanych przez nas struktur danych będzie szybszy w dodawaniu, usuwaniu jak i wyszukiwaniu danych.

Rozmiar struktur ustala się podając je w programie, lecz testy odbywają się na **10 000** elementów powtórzone **10** razy.

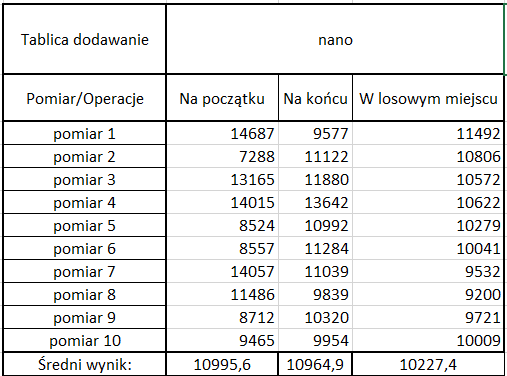
Elementy generowane są za pomocą pętli, w której wartości generują się w sposób uporządkowany (od 0 do (podany rozmiar -1)) lub w sposób losowy.

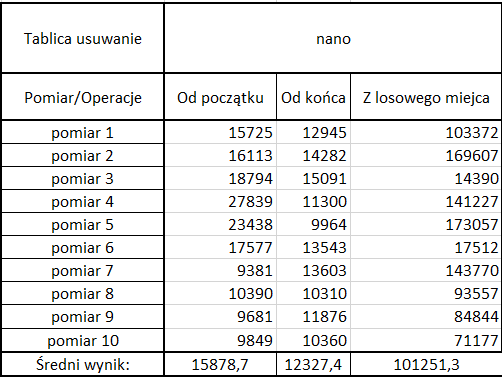
Czas mierzony jest od momentu wykonanej danej funkcji (podany przez nas rozmiar) razy i kończy się zaraz po jej zakończeniu wykonywania.

**Tablica**

Tablica jest to struktura, która pozwala nam na gromadzenie większej ilości danych w uporządkowanej formie.

Dodawanie:

* Najlepszy przypadek:
  + Na początku: O (1)
  + Na końcu: O (1)
  + W losowym miejscu: O(n)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

Usuwanie:

* Najlepszy przypadek:
  + Na początku: O(1)
  + Na końcu: O(1)
  + W losowym miejscu: O(n)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

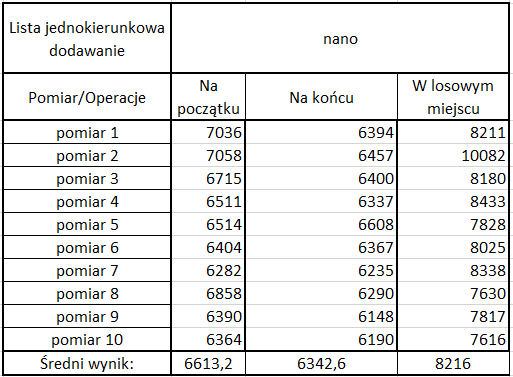
Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieWyszukiwanie:

* + Najlepszy przypadek:  
     O(1)
  + Najgorszy przypadek:  
     O(n)

**Lista jednokierunkowa**

Listajednokierunkowa to struktura danych, która zawiera węzły oraz głowę tablicy, która jest pierwszym węzłem. Każdy z węzłów posiada wartość oraz pole wskazujące na kolejny węzeł. Iterować po niej możemy jedynie w jedną stronę zaczynając od głowy.

Dodawanie:

* Najlepszy przypadek:
  + Na początku: O(1)
  + Na końcu: O(1)
  + W losowym miejscu: O(1)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieUsuwanie:

* Najlepszy przypadek:
  + Na początku: O(1)
  + Na końcu: O(1)
  + W losowym miejscu: O(1)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

* Wyszukiwanie:
  + Najlepszy przypadek:   
     O(1)
  + Najgorszy przypadek:  
     O(n)

**Lista dwukierunkowa**

Listadwukierunkowa to struktura danych, która zawiera węzły, głowę oraz ogon, które również są węzłami wyznaczającymi początek i koniec tablicy. Każdy z węzłów posiada wartość oraz dwa pola wskazujące. Jedno pole wskazuje na poprzedni węzeł, a drugie na następny. Iterować po niej możemy w dwie strony zaczynając od głowy lub ogona tablicy.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieDodawanie:

* Najlepszy przypadek:
  + Na początku: O(1)
  + Na końcu: O(1)
  + W losowym miejscu: O(n)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznieUsuwanie:

* Najlepszy przypadek:
  + Na początku: O(1)
  + Na końcu: O(1)
  + W losowym miejscu: O(n)
* Najgorszy przypadek:
  + Na początku: O(n)
  + Na końcu: O(n)
  + W losowym miejscu: O(n)

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

* Wyszukiwanie:
  + Najlepszy przypadek:  
     O(1)
  + Najgorszy przypadek:  
     O(n)

**Wyniki eksperymentu:**

Obraz zawierający zrzut ekranu, tekst, linia, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, linia, Wykres

Opis wygenerowany automatycznie

**Podsumowanie:**

Jak można zobaczyć we wszystkich, strukturach, gdzie dodawaliśmy od początku, z końca i w losowym miejscu to, że obie listy są szybsze w dodawaniu niż tablica. Co jest ciekawe, że nasza implementacji jest szybsza od zoptymalizowanej tablicy, która daje nam Java.

W usuwaniu można zobaczyć, że czas we wszystkich trzech strukturach jest podobny, kiedy usuwamy od początku, po tym, kiedy usuwamy od końca to lista dwukierunkowa staje się najwolniejsza, a w usuwaniu w losowym miejscu to dwukierunkowa wychodzi z tym, że jest najszybsza.

W wyszukiwaniu najszybszą strukturą jest tablica i może być spowodowane tym zoptymalizowaniem jak i tym, że lista mogła wcześniej już być posortowana

Wniosek z naszego testu wydajnościowego jest taki, że mniej więcej pokrywa się z podaną przez nas tezę jedyne co się nie sprawdziło było wyszukiwanie. Wynik mógłby być trochę inny, gdyby lista, w której szukaliśmy elementu nie była by posortowana i w takim przypadku lista dwukierunkowa byłaby szybsza.