Badanie efektywności operacji dodawania, usuwania oraz wyszukiwania elementów w  różnych strukturach danych

**Spis treści**

Wstęp\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1

Plan eksperymentu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1

Opis struktur daych\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2

Złożoność obliczeniowa metod\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2

Rezultaty\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3-6

Podsumowanie\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 7

Czego nie udało się zrobić\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

**Wstęp:**

Założeniami projektu jest sprawdzenie jak wpływa ilość danych na czas wykonywania operacji w różnych strukturach takich jak np. tablica, lista albo drzewo. W eksperymencie chodzi o napisanie ręcznie funkcji i sprawdzenie ile zajmują im poszczególne zadania takie jak dodawanie elementów, usuwanie elementów oraz wyszukiwanie elementów. Dane na których operujemy są generowane.

Sensem projektu jest wykazanie która struktura jest najlepsza dla danej ilości danych dla danych operacji, takie doświadczenie może pomóc w przyszłości w doborze struktur.

**Plan eksperymentu:**

- Testowane będą struktury pod kątem czasu wykonywania operacji dodawania, wyszukiwania i usuwania elementów do tablicy

- Rozmiary danych jakie będą używane w testach to 100, 1000, 10 000, 50 000, dane będą generowane losowo.

- Do pomiaru czasu wykorzystywany jest własny timer

**Opis struktury danych:**

Tablica – tablica jest podstawową strukturą danych, która umożliwia przechowywanie elementów o jednorodnym typie w sposób uporządkowany, używając indeksów do odwoływania się do poszczególnych elementów.

**Założenia tablicy:**

-każdy element posiada index

-dostęp za pomocą indexów

-elementy tablicy są przechowywane w sposób ciągły co oznacza że każdy element ma adres w pamięci następnego elementu

-tablica ma stały rozmiar

-typy danych musza być takie same

**Złożoność obliczeniowa metod:**

insertIntoArray(int index, int value): Złożoność czasowa tej metody to O(n) w najgorszym przypadku, gdy indeks jest na początku tablicy, ponieważ wymaga przesunięcia wszystkich elementów tablicy. W przypadku, gdy indeks jest na końcu tablicy, złożoność czasowa to O(1). W przypadku, gdy tablica jest pełna i wymaga rozszerzenia, złożoność czasowa to O(n), ponieważ wymaga skopiowania wszystkich elementów do nowej tablicy.

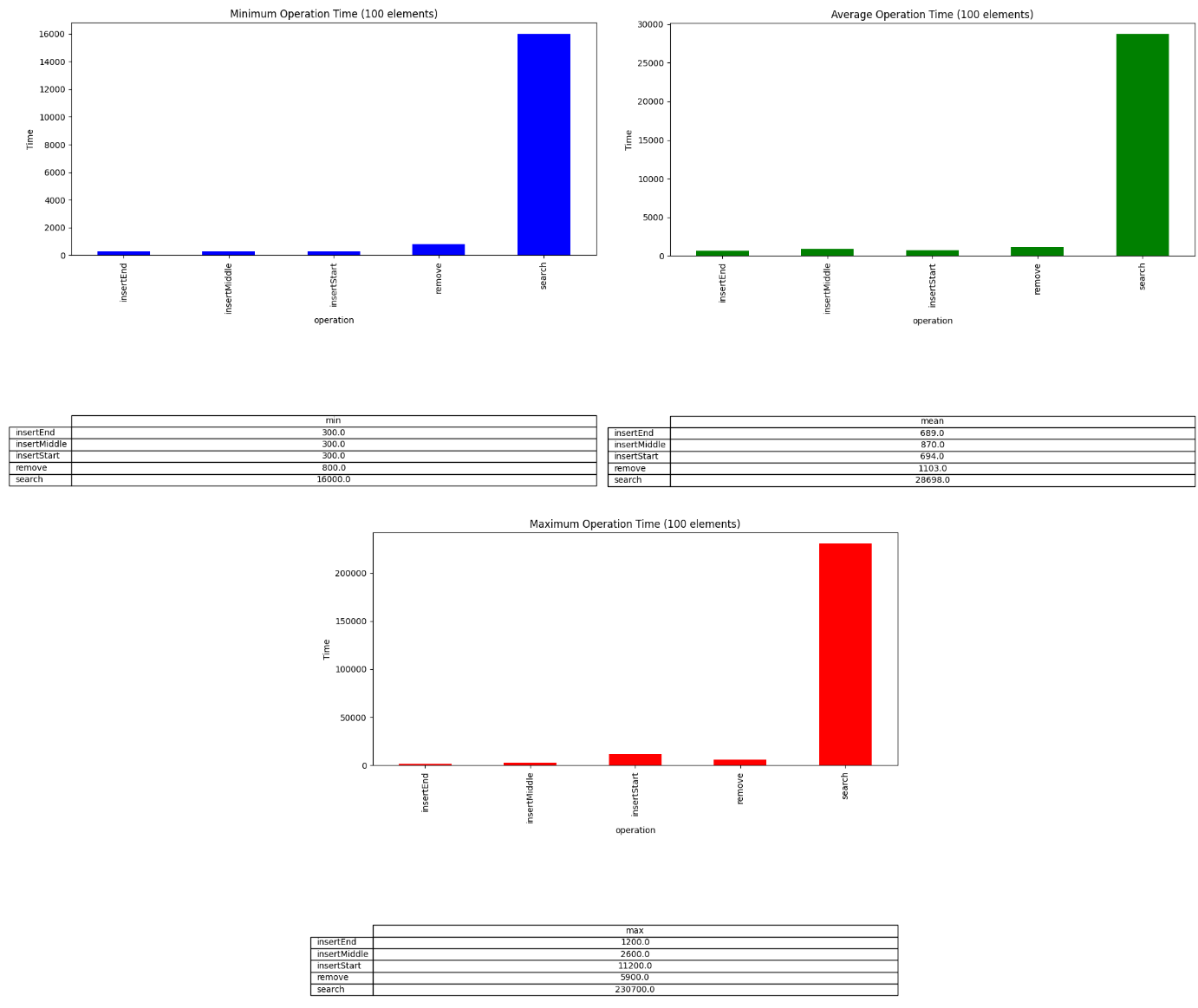
removeByValue(int value): Złożoność czasowa tej metody to O(n) w najgorszym przypadku, gdy wartość jest na początku tablicy, ponieważ wymaga przesunięcia wszystkich elementów tablicy. W przypadku, gdy wartość jest na końcu tablicy, złożoność czasowa to O(1).

search(int value): Złożoność czasowa tej metody to O(n) w najgorszym przypadku, gdy wartość jest na końcu tablicy, ponieważ wymaga przeszukania całej tablicy.

Przedstawienie wyników tablicy 100 elementów:

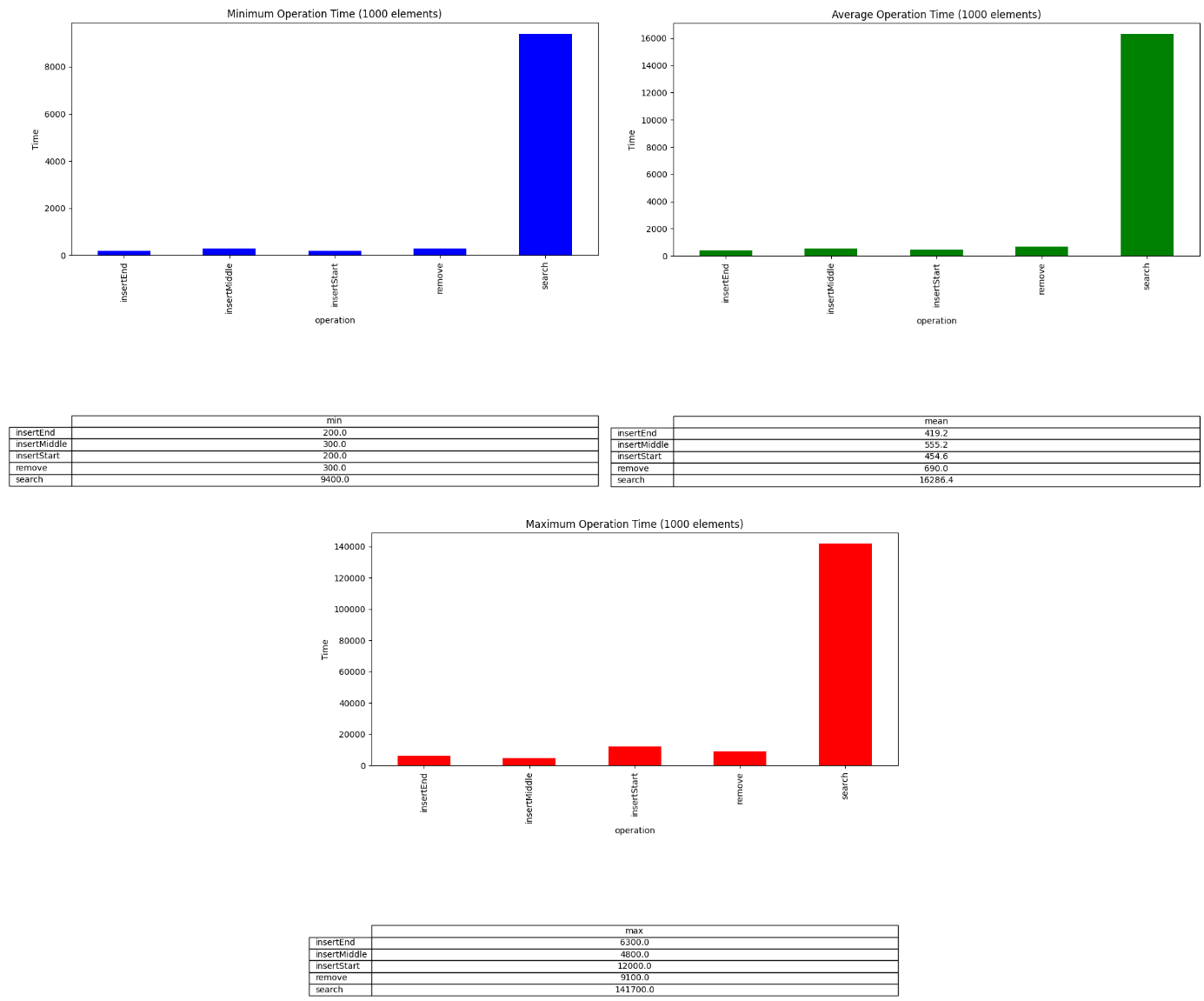
Przy wielkości tablicy 100 zdecydowanie najdłuższą operacją jest wyszukiwanie

danych w tablicy co możemy zobaczeć na wykresie poniżej.



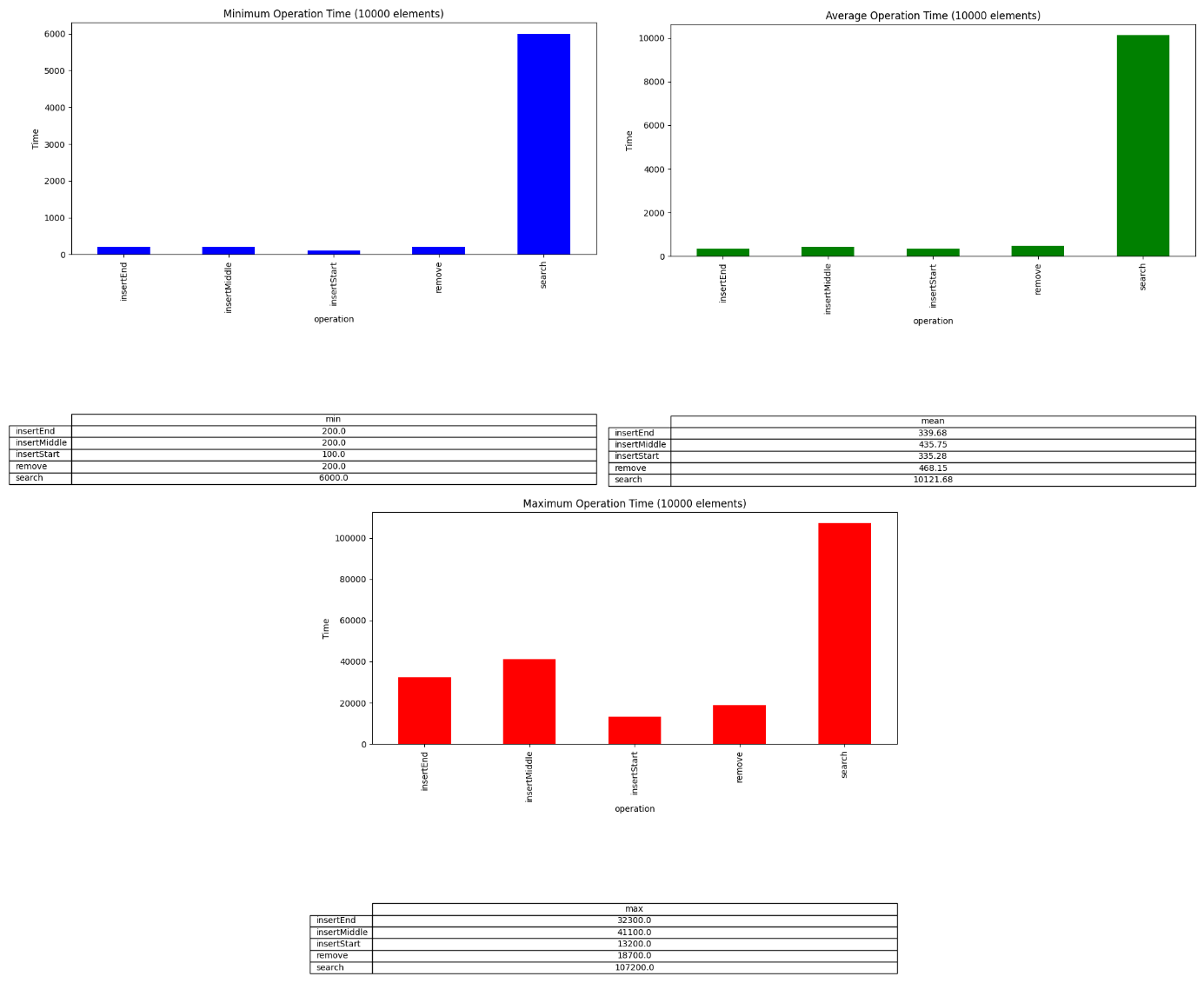
Przedstawienie wyników tablicy 1000 elementów:

Przy wielkości tablicy 1000 elementów widzimy że dalej definitywnie wyszukiwanie jest najdłuższą operacją lecz reszta operacji też lekko się wydłużyła jest to normalny wzrost który jest spowodowany większą ilością elementów.



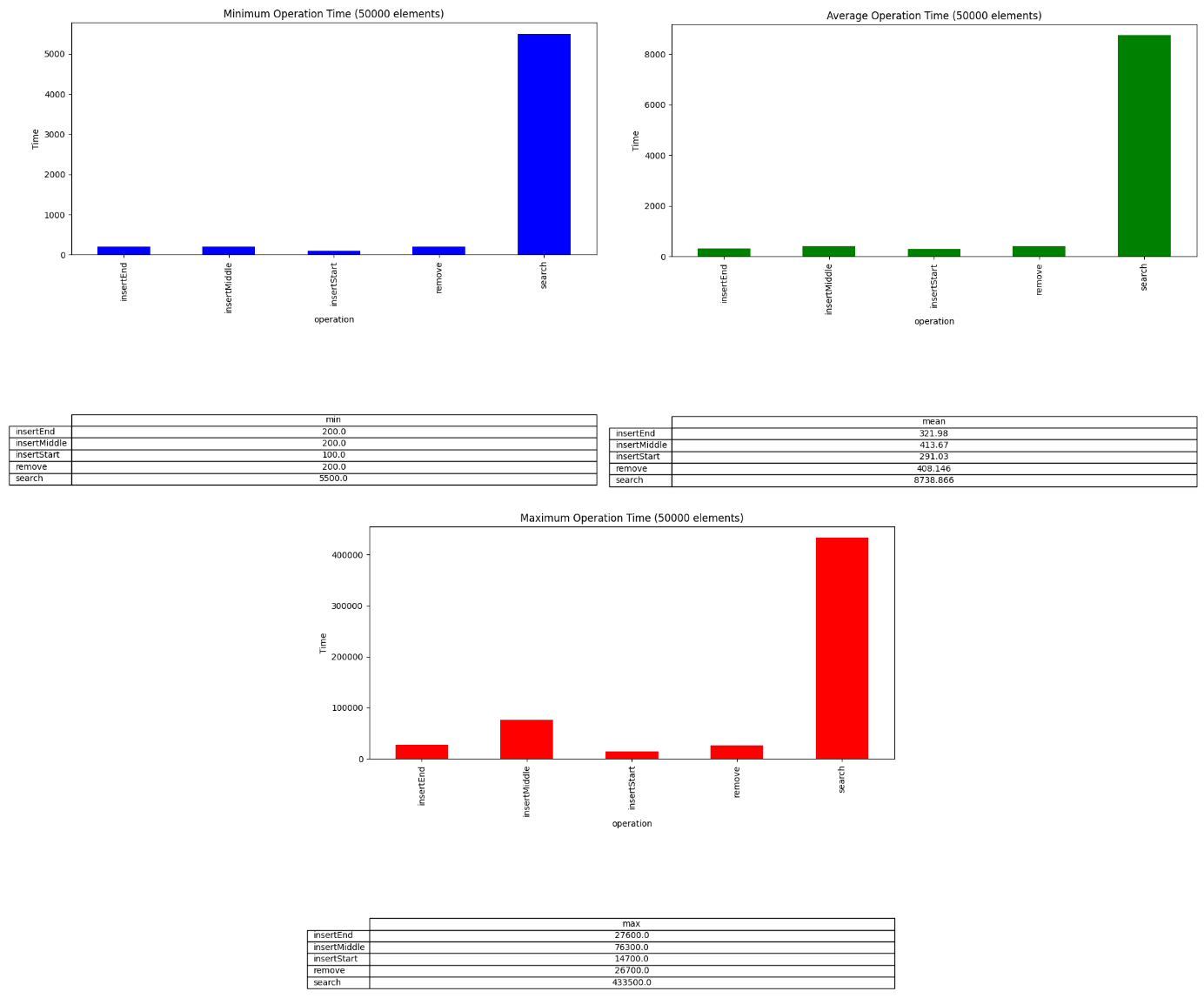
Przedstawienie wyników tablicy 10 000 elementów:

Przy tablicy o wielkości 10 000 możemy zaobserwować ciekawe zjawisko w którym wartości minimalne oraz średnie są bardzo podobne do wartości z tablicy o rozmiarze 1000 elementów ale wartości maksymalne są o wiele większe owszem dalej wyszukiwanie jest najdłuższym procesem ale reszta operacji też zwiększyła znacząco swoje czasy.



Przedstawienie wyników tablicy 50 000 elementów:

Przy tablicy o wielkości 50 000 możemy zaobserwować spadek maksymalnego czasu wykonywania wszystkich operacji oprócz wyszukiwania.



**Podsumowanie:**

Wnioski jaki możemy wyciągnąć z tych danych są takie że operacja wyszukiwania jest najbardziej czasochłonną co za tym idzie obciążającą operacja z tych podanych jeśli mowa o tablicy elementów. Jest to o tyle ważne że jeśli chcielibyśmy coś pisać w oparciu o tablicę danych to musimy pamiętać że operacja wyszukiwania jest najbardziej czasochłonna i obciążająca co może być ważne w kontekście urządzenia na którym będzie to działać. Ten eksperyment pokazuje że do tablica może nie być idealna w każdym zastosowaniu.

**Czego nie udało się zrobić:**

Niestety nie udało się wykonać innych rodzajów struktur danych oprócz tablicy.

Również nie udało się zrobić usuwania i wyszukiwania na początku środku i końcu.