

Złożone struktury danych

Maciej Kaszkowiak, 151856

1 METODOLOGIA POMIARU WYDAJNOŚCI STRUKTUR DANYCH

Zmierzyłem czas wykonywania wybranych operacji dla następujących struktur danych:

- Binary Search Tree
- AVL Tree

Zmierzone operacje dla drzew AVL i BST obejmowały:

- tworzenie struktury danych
- wyszukanie najmniejszego elementu
- wypisanie elementów in-order

Ponadto zmierzyłem czas równoważenia elementów w drzewie BST za pomocą algorytmu Day Stout Warren.

Algorytmy zostały zaimplementowane w języku Python 3.8.10. Testy zostały uruchomione pod systemem Ubuntu 20.04.2 działającym na platformie WSL2.

Dane wejściowe zostały wygenerowane w następujący sposób:

- Losowy ciąg liczb z przedziału $<-N, N>$ bez duplikatów

Czas został zmierzony dla siedemnastu różnych rozmiarów danych wejściowych, gdzie liczba elementów to następująco: 4, 7, 12, 21, 37, 64, 111, 194, 338, 588, 1024, 1783, 3104, 5405, 9410, 16384, 28526.

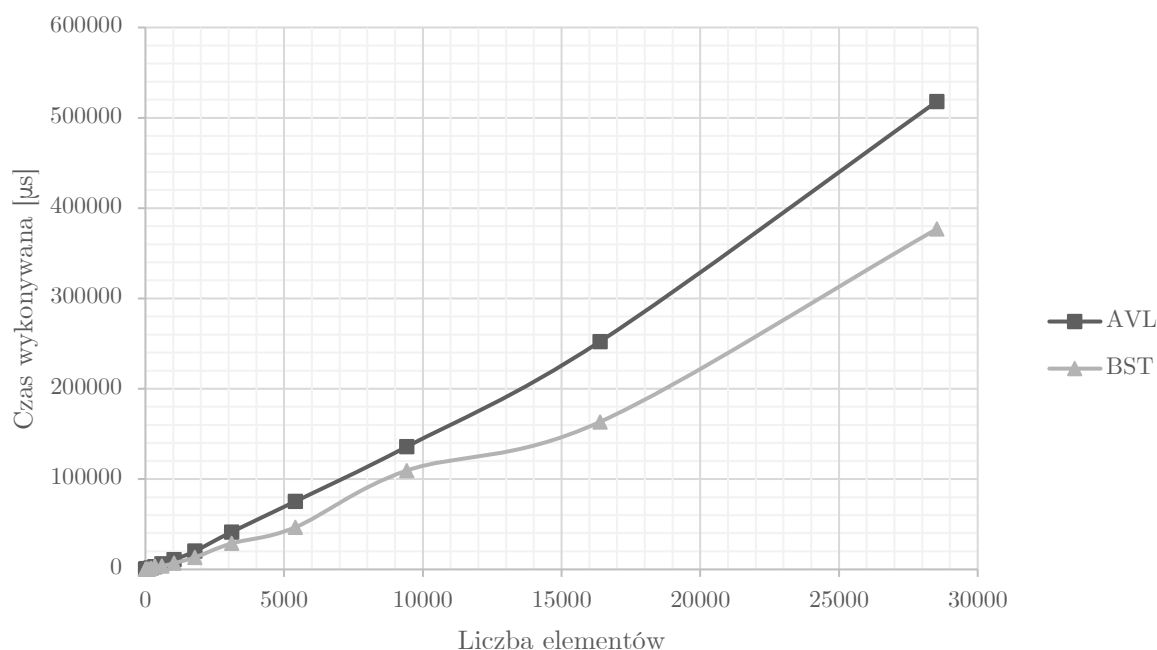
Pomiary zostały wykonane z dokładnością do mikrosekund. Pominęto czas generowania danych wejściowych.

2 WYDAJNOŚĆ STRUKTUR DANYCH

2.1 TWORZENIE DRZEW A AVL I BST

Drzewo BST zostało utworzone za pomocą algorytmu wyszukiwania połówkowego. Dane wejściowe zostały posortowane, a następnie dodano środkowe elementy kolejnych części tablicy. Sortowanie zostało celowo pominięte w pomiarze czasu. Złożoność czasowa tworzenia drzewa BST to $O(n \log n)$, natomiast ze względu na pominięte sortowanie zmierzony czas sprowadza się do $O(n)$.

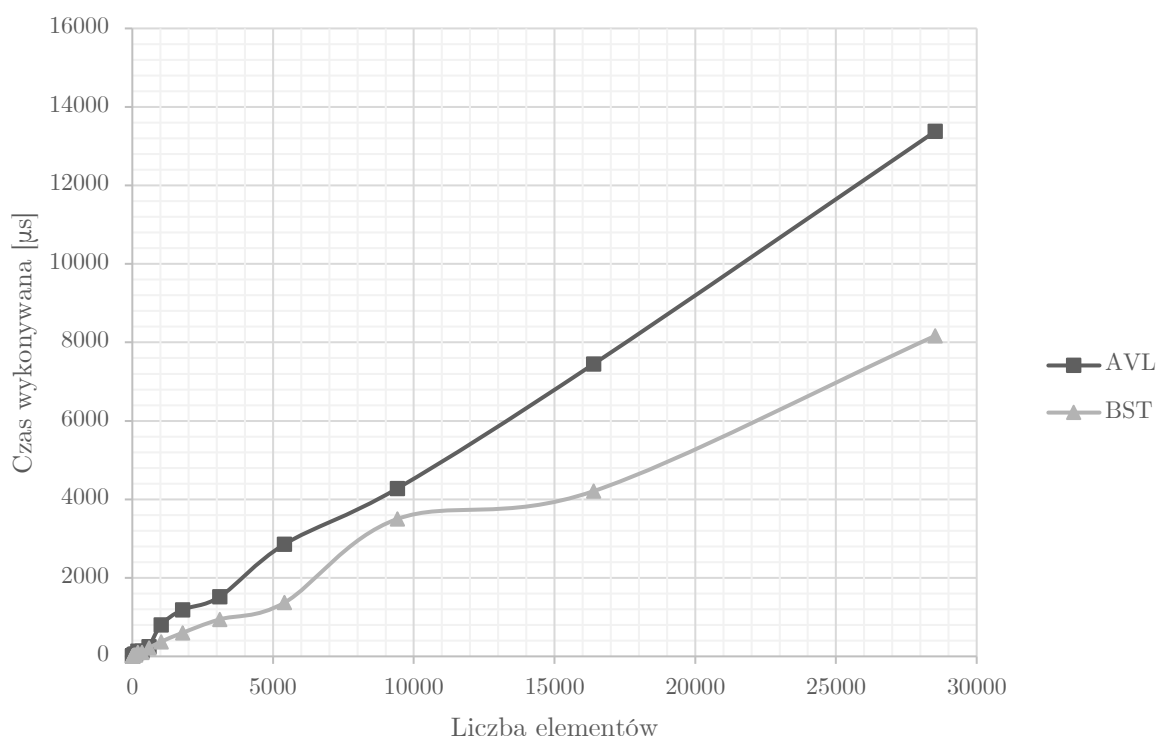
Drzewo AVL zostało utworzone na bazie nieposortowanych danych wejściowych. Drzewo wykonało procedurę balansowania przy każdym dodawanym elemencie który powodował niezrównoważenie drzewa. Złożoność czasowa tworzenia drzewa AVL to $O(n \log n)$.



2.2 WYPISANIE ELEMENTÓW IN-ORDER W DRZEWACH AVL I BST

Wypisanie elementów in-order w drzewach AVL oraz BST polega na rekursywnym odczytywaniu kolejnych elementów w głąb drzewa. Złożoność czasowa procedury wynosi $O(n)$.

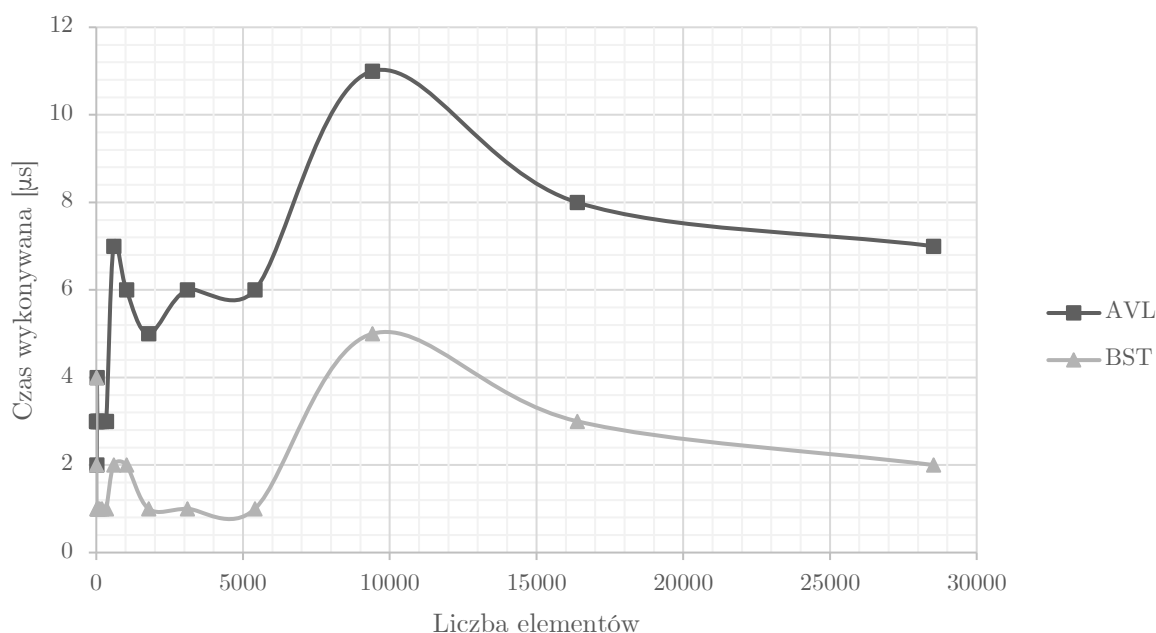
Na wykresie możemy zauważyć rozbieżność czasową wydajności procedury pomiędzy drzewem AVL a drzewem BST. Pomiar pozornie nie zgadza się z teorią, ponieważ oba drzewa opierają się na tej samej implementacji struktury drzewa oraz procedury in-order. Powodem jest odmienna struktura drzewa wywołana innymi sposobami tworzenia struktury. Dla identycznego zestawu danych, drzewo AVL charakteryzuje się minimalnie większą wysokością od drzewa BST. Skutkuje to większą głębokością rekursji w procedurze in-order dla drzewa AVL i tym samym większym czasem wykonywania. Można postawić tezę, że drzewo BST utworzone metodą wyszukiwania połówkowego jest bliższe do idealnej reprezentacji.



2.3 ZNALEZIENIE MINIMALNEGO ELEMENTU W DRZEWACH AVL I BST

Znalezienie minimalnego elementu w drzewach AVL i BST to procedura ze złożonością czasową $O(\log n)$. Algorytm polega na przemierzaniu lewej części drzewa począwszy od korzenia, aż do napotkania skrajnie lewostronnego liścia. Liczba wywołań procedury zależy tym samym od wysokości drzewa.

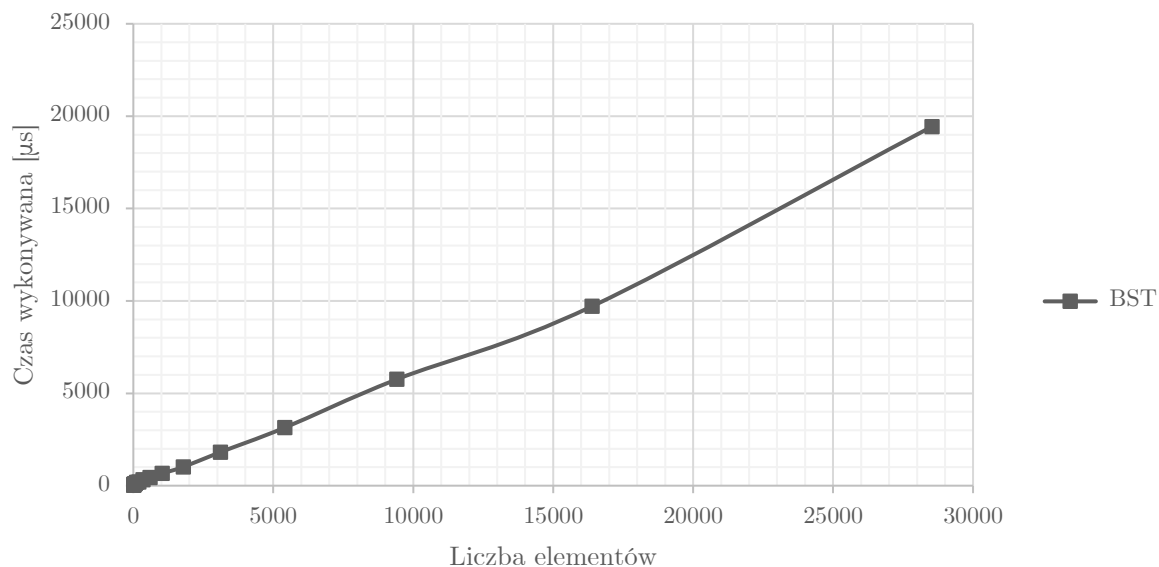
Możemy zaobserwować, że jest to bardzo optymalna procedura. Wykres zależności czasu od liczby elementów nie odwzorowuje prawdziwego kształtu krzywej ze względu na skalę pomiarów. Opóźnienie rzędu 5 mikrosekund może z łatwością zostać wywołane przez chwilowe obciążenie procesora. Ponadto drzewo AVL wydaje się być powtarzalnie wolniejsze od drzewa BST. Różnica jest wywołana sposobem implementacji pomiaru algorytmu. Interpreter prawdopodobnie optymalizuje kod, który wykorzystuje te same zmienne do pomiaru najpierw drzewa AVL, a następnie drzewa BST.



2.4 RÓWNOWAŻENIE DRZEWA BST

Równoważenie drzewa BST zostało zaimplementowane za pomocą algorytmu Day Stout Warren. Pomiar został wykonany poprzez wygenerowanie ciągu losowych liczb, utworzenie drzewa BST poprzez iteracyjne wstawianie kolejnych elementów, a następnie zmierzenie czasu potrzebnego na zrównoważenie drzewa.

Algorytm Day Stout Warren posiada złożoność obliczeniową $O(n)$. Polega na zamianę oryginalnego drzewa w jego zdegenerowaną postać, a następnie zrównoważenie winorośli za pomocą odpowiednich rotacji.



3 WNIOSKI

Drzewa AVL oraz BST to wydajne struktury danych stanowiące doskonałą alternatywę do operowania na posortowanej liście danych. Wybór odpowiedniego drzewa powinien zależeć od docelowych wykonywanych operacji. Stworzenie drzewa BST poprzez wyszukiwanie połówkowe może zapewnić możliwie najmniejszy czas dostępu do danych, ze względu na optymalną wysokość drzewa. Przewidzenie regularnej modyfikacji węzłów może przechylić szalę na korzyść drzewa AVL, które automatycznie wywołuje optymalną procedurę równoważenia.

4 SPIS TREŚCI

1	Metodologia pomiaru wydajności struktur danych.....	1
2	Wydajność struktur danych.....	2
2.1	Tworzenie drzewa AVL i BST.....	2
2.2	Wypisanie elementów in-order w drzewach AVL i BST.....	3
2.3	Znalezienie minimalnego elementu w drzewach AVL i BST.....	4
2.4	Równoważenie drzewa BST.....	5
3	Wnioski	6