Sprawozdanie – Drzewa Binarne

Kacper Leporowski
Adam Malinowski

W tym sprawozdaniu opisane i porównane pod względem złożoności i czasu wykonania będą pewne operacje wykonywane na dwóch rodzajach drzew binarnych – drzewie BST (Binary Search Tree, drzewo przeszukiwania binarnego) oraz drzewie AVL, które jest zrównoważonym drzewem BST.
Do przeprowadzenia testów wykorzystano wcześniej wygenerowane ciągi liczbowe – od dwóch do dwudziestu tysięcy elementów, z krokiem 2000. Ciągi były posortowane rosnąco.

Tworzenie drzewa

W przypadku drzewa BST, złożoność tworzenia drzewa wynosi $O(n^2)$. Jest to spowodowane tym, że dane wejściowe są posortowane rosnąco i wstawiane do drzewa po kolei, co skutkuje powstaniem drzewa zdegenerowanego (winorośli). W tym przypadku dla każdego węzła drzewa posiada on jedynie prawego potomka, a to powoduje, że wstawienie elementu ma złożoność O(n).
Dla algorytmu AVL tworzenie drzewa ma złożoność O(n logn) – wstawianie elementu ma złożoność O(logn), ponieważ taka jest wysokość drzewa. Ogólnie dla drzew BST/AVL złożoność operacji wstawiania jest równa wysokości drzewa, a ta zależy od danych wejściowych i tego, czy drzewo jest zrównoważone (stąd niższa złożoność dla AVL w tym konkretnym przypadku).

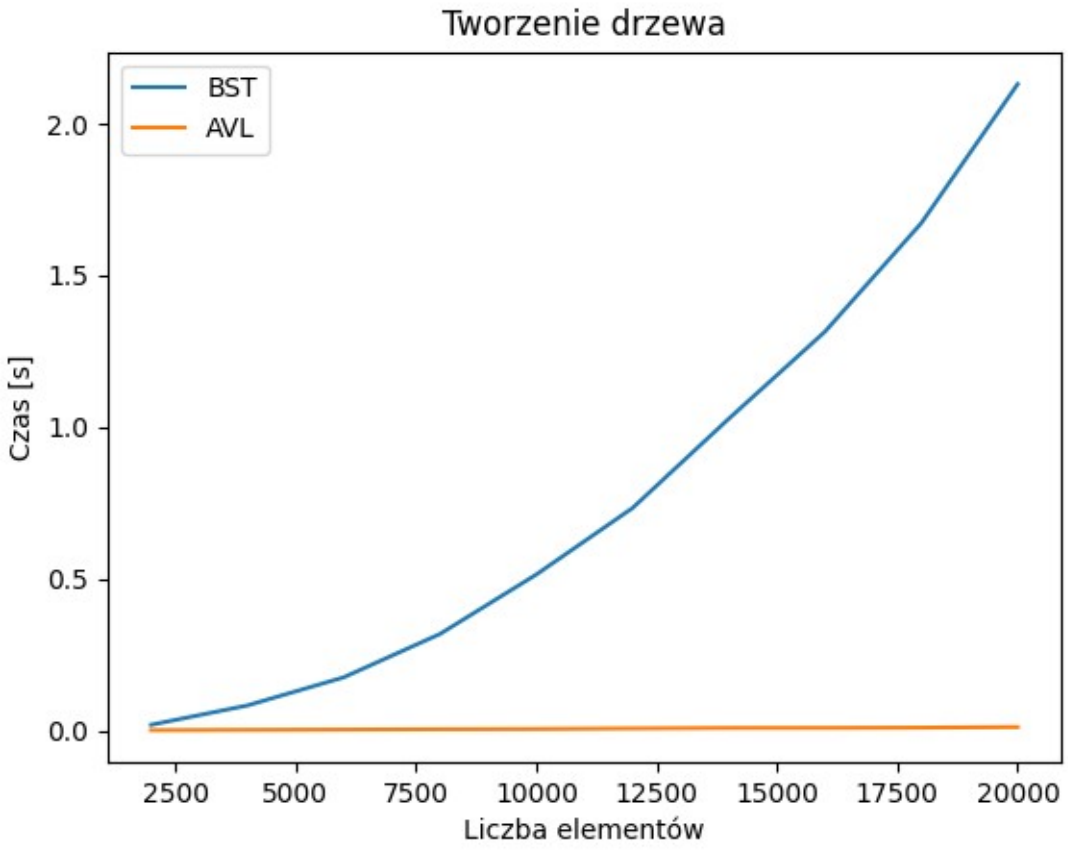

Figura 1: Czas wykonania - tworzenie

Wyszukiwanie maksimum

Wyszukiwanie elementu o największej wartości polega na przeszukiwaniu kolejnych prawych potomków korzenia aż do ostatniego. Operacja ta również jest zależna od wysokości drzewa, stąd dla zrównoważonego drzewa AVL ma ono średnią złożoność O(log n), a dla zdegenerowanego drzewa BST – O(n).

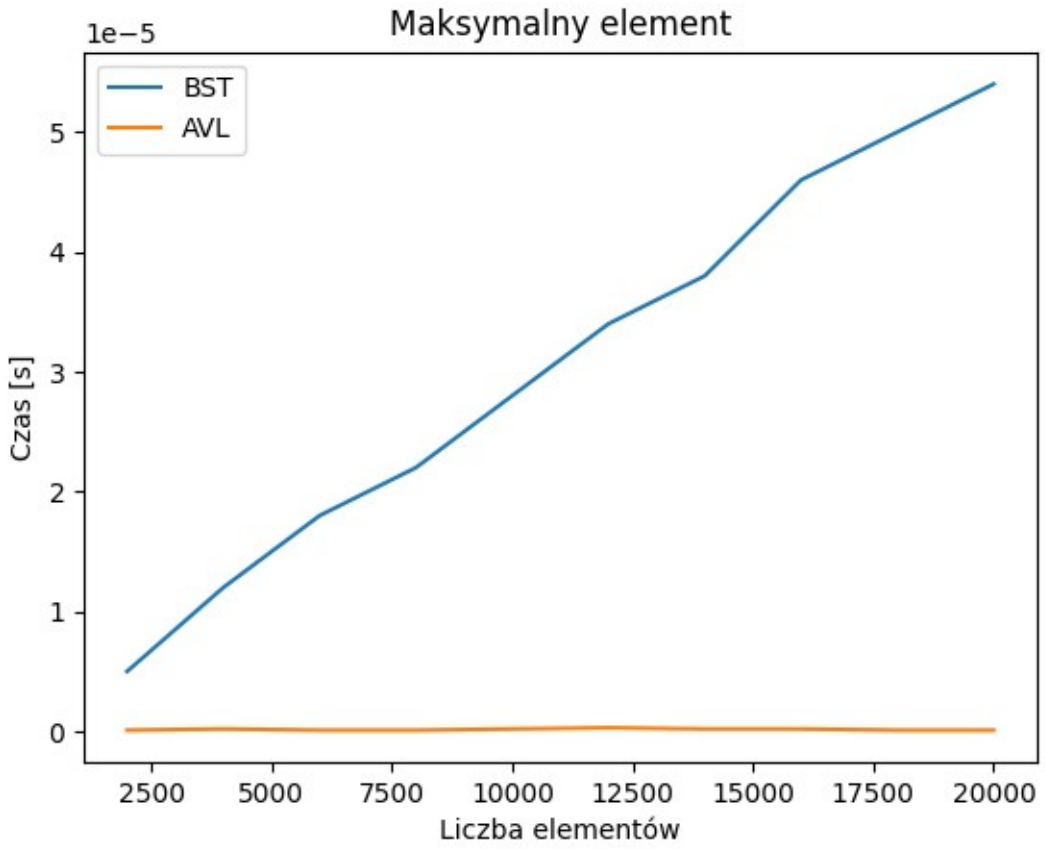

Figura 2: Czas wykonania - maksimum

Wypisanie in-order

Wypisywanie wszystkich elementów w kolejności in-order wykorzystuje algorytm przechodzenia przez drzewo w taki sposób, że dla danego węzła najpierw odwiedzany jest jego lewy potomek, potem korzeń, a potem prawy potomek. Ogólna złożoność wynosi O(n) – każdy z węzłów zostanie odwiedzony przynajmniej raz.
W tym przypadku algorytm wykonuje się szybciej dla drzewa BST – ponieważ jest ono w postaci zdegenerowanej, każdy z węzłów zostanie odwiedzony dokładnie raz – w drzewie AVL dojdzie do sytuacji, w której algorytm przejdzie przez dany węzeł kilka razy.

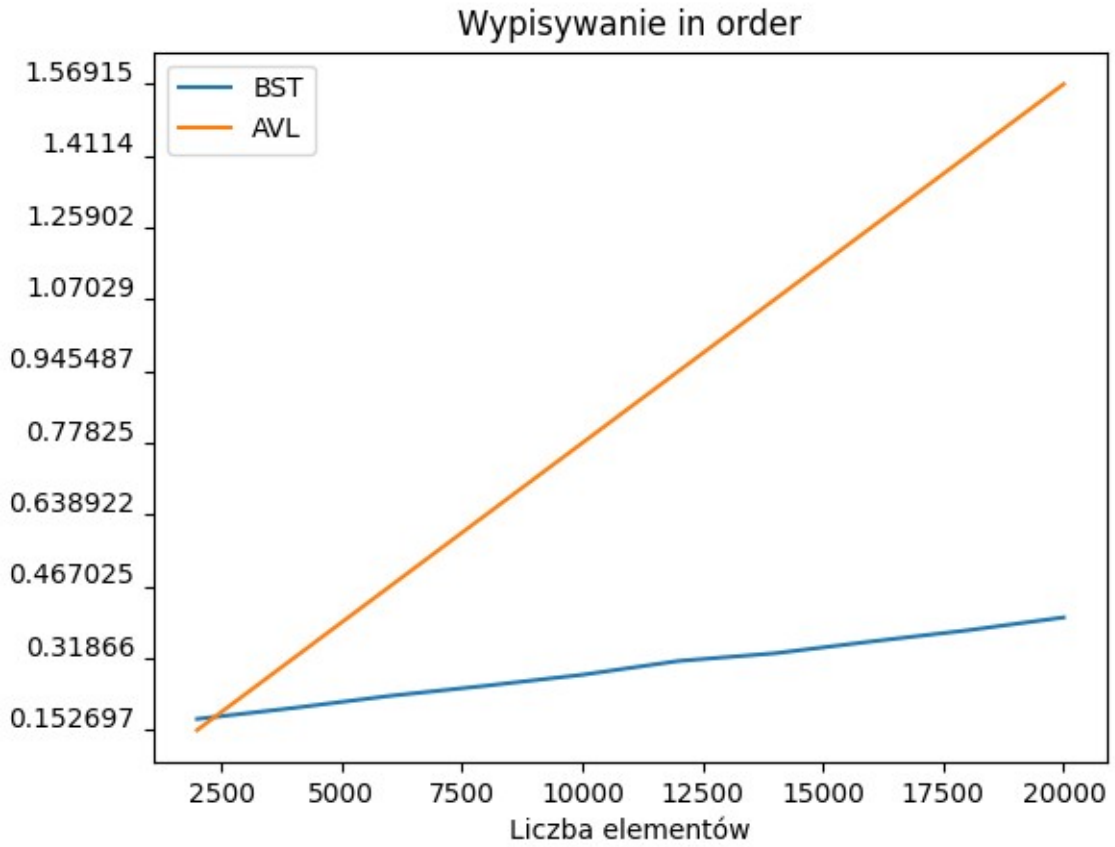

Figura 3: Czas wykonania - wypisywanie in-order

Równoważenie drzewa BST

Do równoważenia drzewa BST wykorzystany został algorytm DSW (od nazwisk autorów – Day-Stout-Warren). W pierwszej kolejności algorytm dokonuje na drzewie rotacji w celu otrzymania winorośli – w prezentowanym przypadku drzewo już jest w postaci zdegenerowanej, więc krok ten jest pomijany. Następnie na podstawie liczby węzłów obliczana jest wysokość drzewa zrównoważonego i wykonywana jest odpowiednia liczba rotacji, a ostatecznie algorytm ma złożoność liniową O(n).

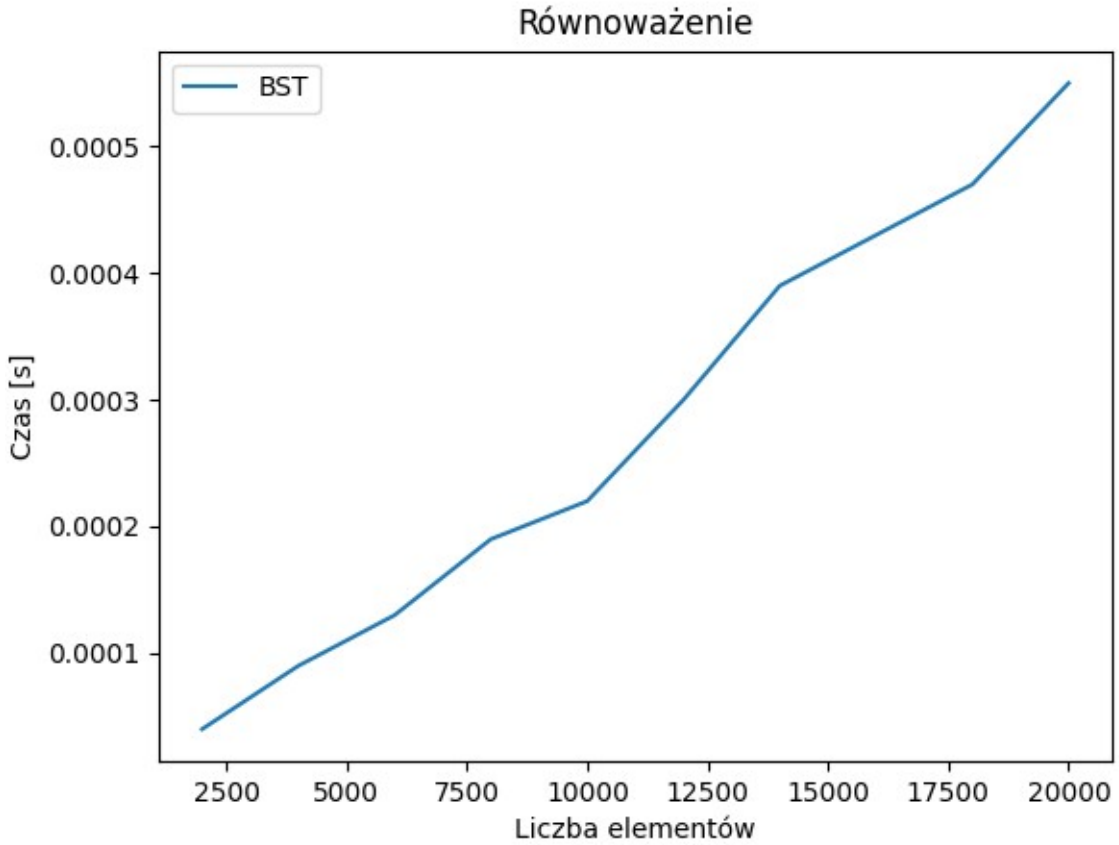

Figura 4: Czas wykonania - DSW