

AISDI – laboratorium z drzew

Jakub Stępniaak
Przemysław Wyziński
Lab 102, grupa 4

Cel ćwiczenia

Celem ćwiczenia była implementacja i testowanie 2 drzew:

- Drzewo BST
- Drzewo AVL

Posiadają one następujące opcje:

- Wstawianie elementu do drzewa (metoda insert)
- Wyszukiwanie elementu w drzewie (metoda get)
- Usuwanie elementu z drzewa (metoda delete)
- Wyświetlanie drzewa w konsoli (metoda display)

Testy drzew

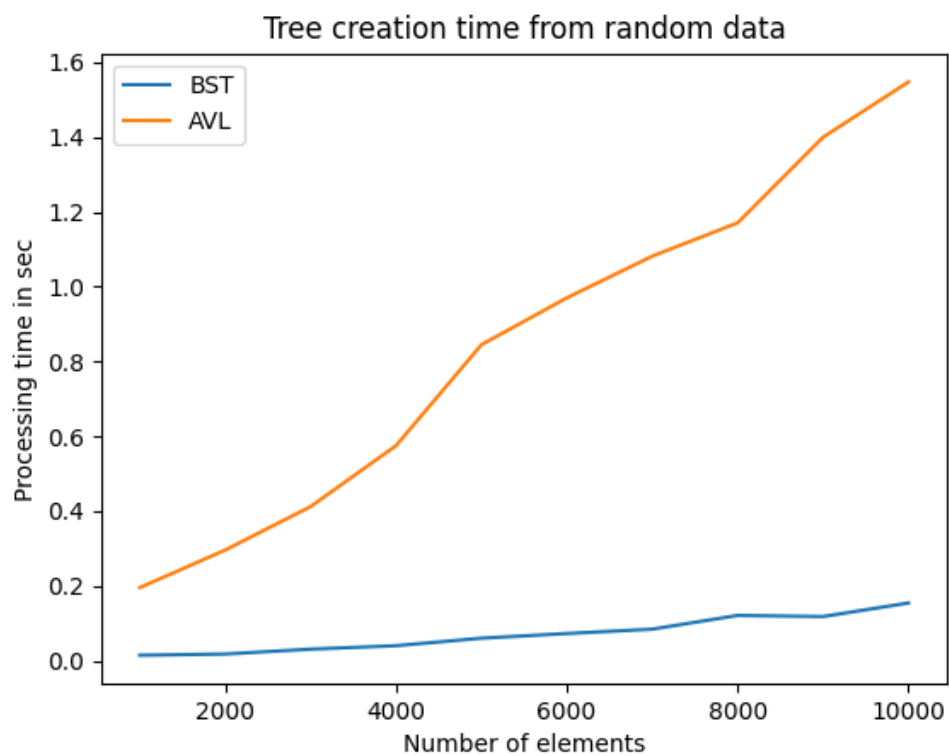
Wszystkie testy zostały wykonane dla dwóch rodzajów argumentów:

- Lista losowych liczb naturalnych ustawionych w losowej kolejności,
- Lista losowych liczb naturalnych ustawionych w kolejności rosnącej.

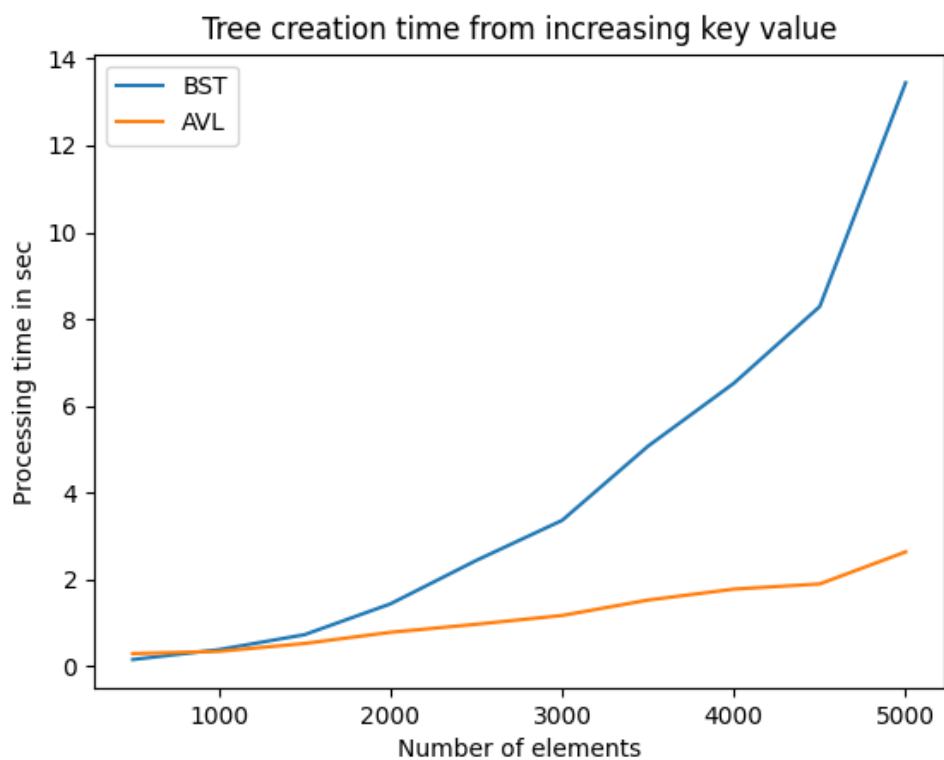
Tworzenie drzew polegało na dodawaniu kolejnych elementów z listy.

Czas potrzebny na utworzenie drzewa

Wyniki dla listy losowych liczb naturalnych ustawionych w losowej kolejności:



Wyniki dla listy losowych liczb naturalnych ustawionych w kolejności rosnącej:



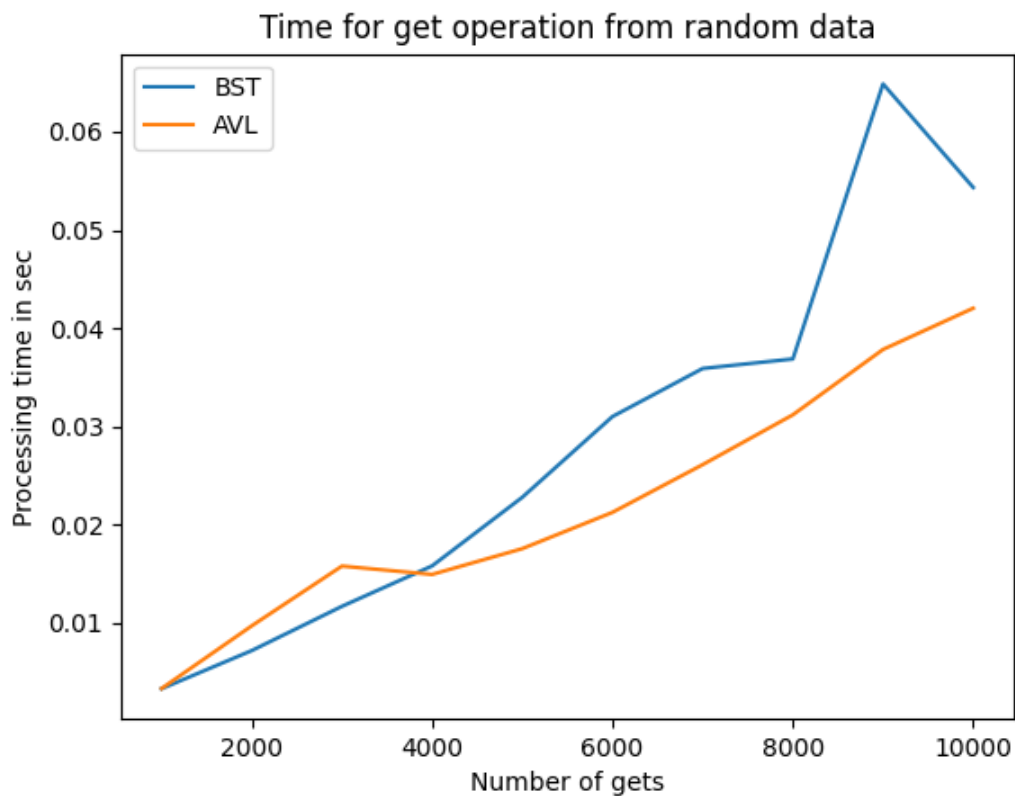
Wnioski:

Dla przypadku ogólnego tworzenie drzewa BST okazuje się szybsze. Powodem jest fakt, że AVL musi się reorganizować, aby zapewnić balans wysokości.

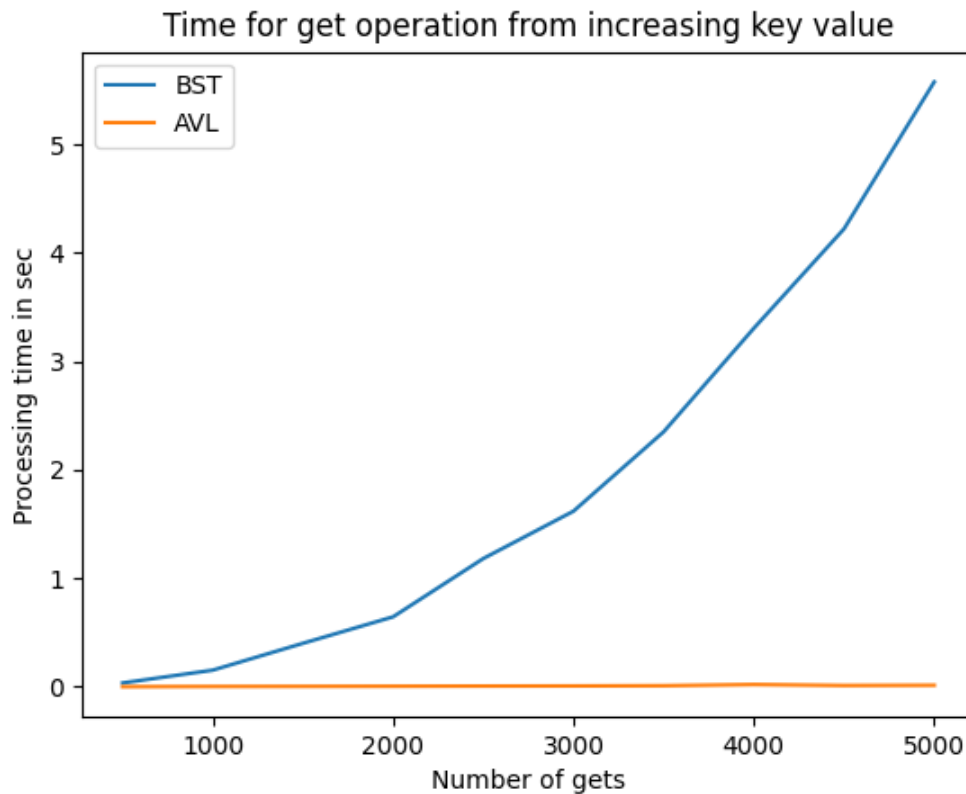
Inaczej jest jednak dla przypadku pesymistycznego tworzenia drzewa BST. Gdy dodajemy kolejne klucze w kolejności rosnącej, to drzewo BST rozgałęzia się tylko w jedną stronę, co powoduje, że wysokość drzewa bardzo szybko rośnie, co koreluje ze znacznym wzrostem czasu potrzebnego na utworzenie drzewa.

Czasy potrzebne na wyszukanie wartości przy pomocy klucza:

Wyniki dla listy losowych liczb naturalnych ustawionych w losowej kolejności:



Wyniki dla listy losowych liczb naturalnych ustawionych w kolejności rosnącej:



Wnioski:

Dla przypadku losowego czasy są podobne, ponieważ wyszukujemy pierwsze klucze, które utworzyły drzewo. Modyfikacja AVL nie poprawia znacznie czasów wyszukiwania.

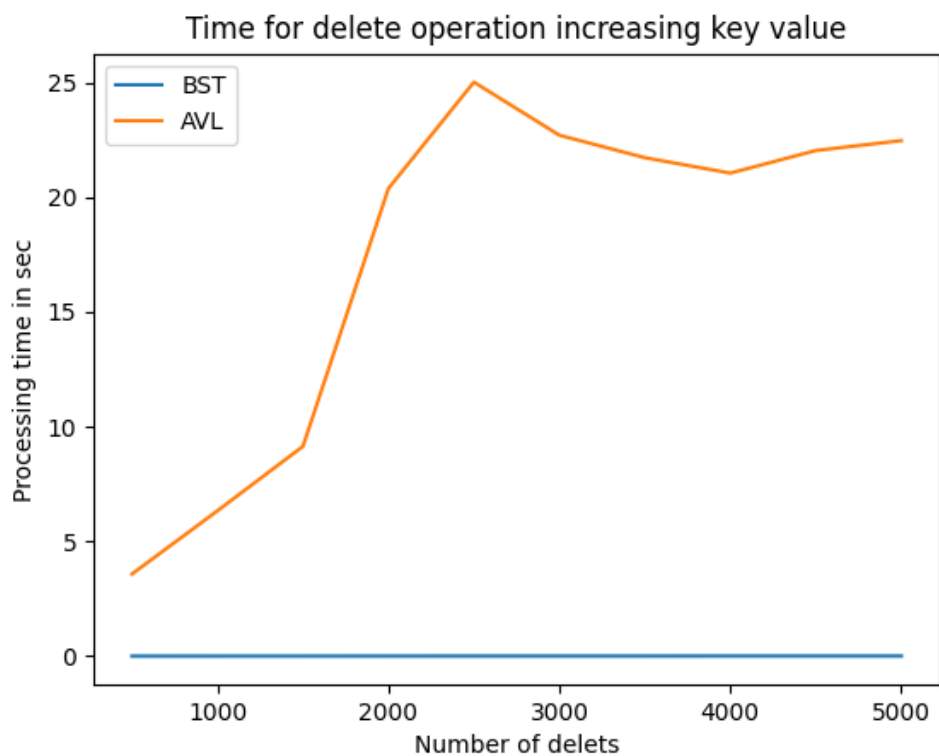
Dla przypadku kluczy w kolejności rosnącej BST znacznie gorzej radzi sobie z wyszukiwaniem kluczy. Wiąże się to z koniecznością przechodzenia po drzewie bardzo głęboko. Drzewo AVL zmniejsza swoją wysokość, przez co wyszukiwanie w takim drzewie jest znacznie szybsze.

Czasy potrzebne na usunięcie klucza:

Wyniki dla listy losowych liczb naturalnych ustawionych w losowej kolejności:



Wyniki dla listy losowych liczb naturalnych ustawionych w kolejności rosnącej:



Wnioski:

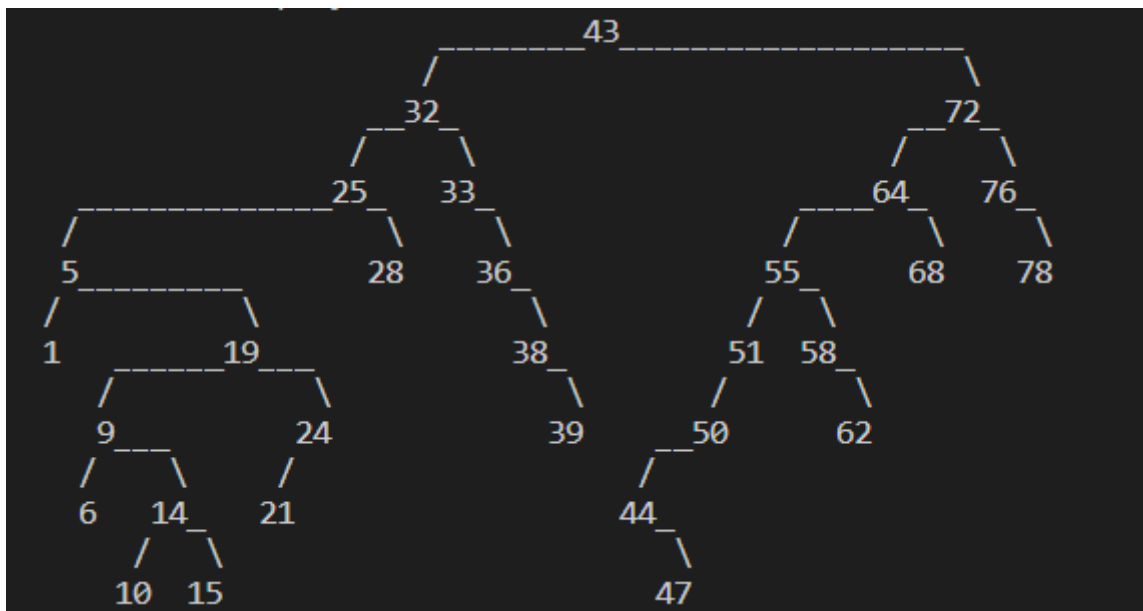
W obu przypadkach usuwanie w drzewie AVL okazuje się wolniejsze. Jest to spowodowane tym, że za każdym razem drzewo musi sprawdzić, czy nie został zaburzony balans wysokości, a gdy wystąpi musi się zreorganizować.

Podsumowanie testów:

Samo przygotowanie drzewa AVL oraz jego modyfikacje zajmują więcej czasu niż analogiczne operacje dla drzewa BST. Jednak samo wyszukiwanie dla drzewa AVL jest znacznie szybsze niż dla BST.

Wyświetlanie drzew za pomocą metody display:

Drzewo BST:



Drzewo AVL:

