

AISDI 2024L

Projekt 3

Drzewa

Maciej Bogusławski 331362

Hubert Kaczyński 331386

Opis projektu

W pliku BST.py znajduje się implementacja klasy BST_Node, która pozwala na tworzenie węzłów drzewa BST oraz klasy BST_Tree, która odpowiada za strukturę drzewa, wstawianie, wyszukiwanie i usuwanie elementów.

W pliku AVL.py znajduje się implementacja klasy AVL_Node, która pozwala na tworzenie węzłów drzewa AVL oraz klasy AVL_Tree, która odpowiada za strukturę drzewa, wstawianie, wyszukiwanie elementów i równoważenie drzewa poprzez odpowiednie rotacje.

W pliku test_bst.py i test_avl.py znajdują się testy jednostkowe sprawdzające poprawność działania metod wyżej wymienionych klas.

W pliku errors.py znajduje się implementacja specjalnych wyjątków.

W pliku analyze.py znajduje się implementacja funkcji wywołujących prezentację rozwiązania i odpowiednie pomiary czasowe.

W pliku graph.py znajduje się implementacja przeniesienia wyników pomiarów czasowych na postać graficzną.

W pliku main.py znajduje się funkcja main, wywołująca prezentację rozwiązania.

W folderze results w pliku result.png znajduje się wygenerowany wykres zależności czasu wykonywania operacji wstawiania, wyszukiwania i usuwania węzłów od ilości danych.

W folderze results w plikach BST_drawing.png i AVL_drawing.png znajdują się zrzuty ekranu z przykładami wyświetlania drzew obu rodzajów.

Opis środowiska

Implementacja i testowanie kodu przeprowadzone zostało, wykorzystując wersję 3.10.12 64-bit języka Python.

Projekt korzysta z następujących bibliotek zewnętrznych:

- matplotlib 3.8.3
- pytest 8.1.1

Oraz następujących bibliotek wewnętrznych:

- gc
- time
- random

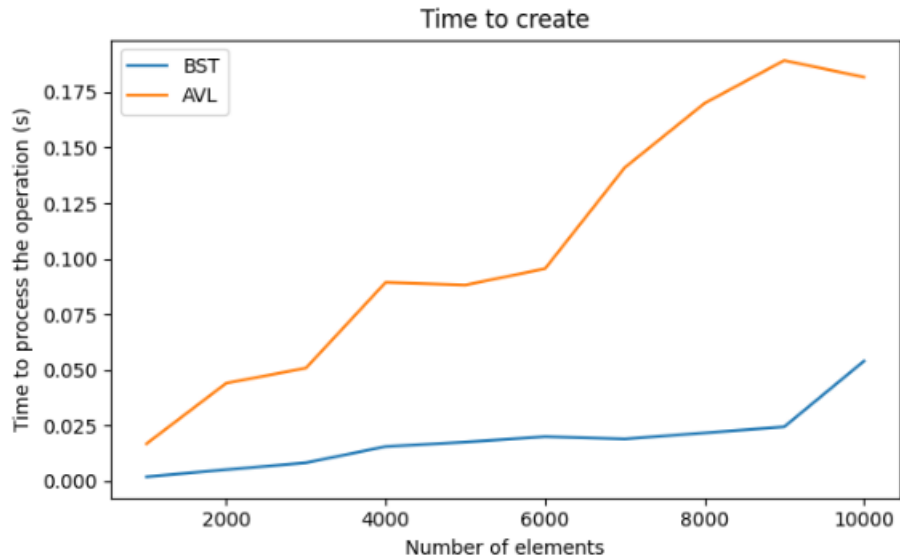
Procedura uruchomienia

W celu poprawnego działania projekt należy uruchomić poprzez plik main.py z głównego folderu repozytorium, wpisując w terminal „python3 Drzewa/main.py”. Jeśli drzewa wyświetlają się w terminalu w sposób niepoprawny ze względu na włączone zawijanie wierszy, należy użyć w terminalu skrótu Alt+Z i uruchomić program ponownie.

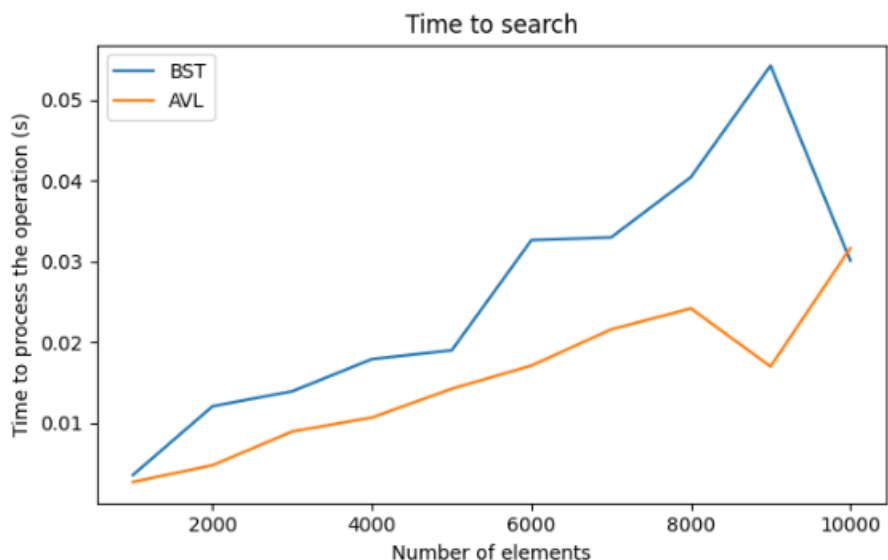
Podział zadań

Cała zawartość projektu była przez nas współtworzona na każdym etapie opracowywania, implementacji, refaktoryzacji oraz testowania rozwiązania.

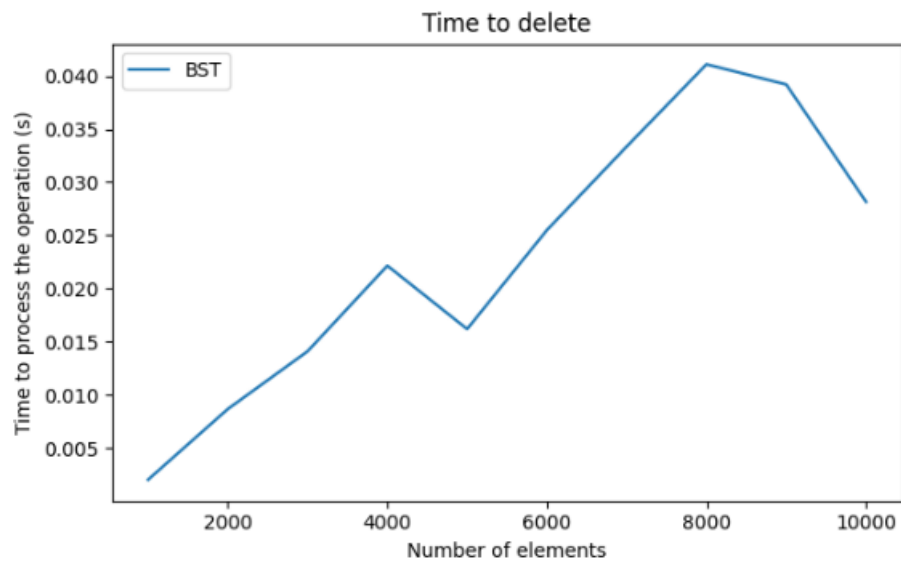
Podsumowanie wyników



Powyższy wykres zgadza się z naszym pierwotnymi przewidywaniami. Drzewo AVL potrzebuje więcej czasu na wstawienie elementów ze względu na potrzebę każdorazowego równoważenia drzewa przy wstawianiu elementu. W wypadku BST, takowe równoważenie nie występuje – przekłada się to na krótszy czas wykonywania operacji.

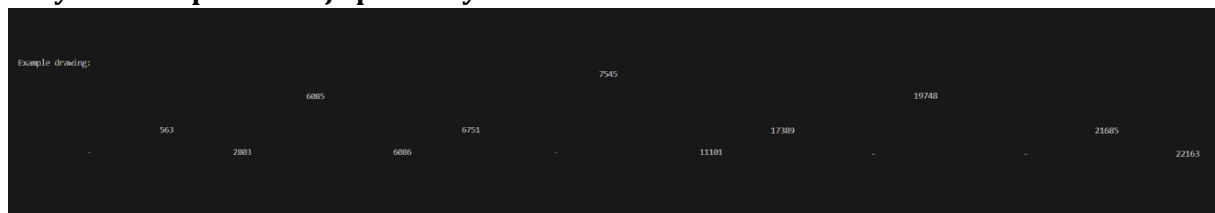


W przypadku wyszukiwania węzła w drzewie, drzewo BST, ze względu na swoją niezrównoważoną budowę wymaga więcej czasu od AVL (poddrzewa drzewa BST są na ogół dłuższe).



Wykres zależności czasu usuwania wszystkich węzłów z drzewa BST od ilości elementów w drzewie jasno wykazuje trend rosnący. Zgodnie z poleceniem, usuwania w drzewie AVL nie zaimplementowano.

Przykładowa prezentacja pierwszych 20 elementów drzewa BST



Przykładowa prezentacja pierwszych 20 elementów drzewa AVL

