

Politechnika Warszawska
Algorytmy i struktury danych

Laboratorium 4
Projekt "Drzewa"
Informatyka – Inteligentne systemy

Paweł Sarnacki 305290

Piotr Niedziałek 304474

Prowadzący: dr inż. Łukasz Skonieczny

Warszawa 2023

1. Wstęp

Program został napisany w języku python z wykorzystaniem aplikacji Visual Studio Code. W programie tworzone są drzewa BST oraz AVL dla 'n' losowych liczb od 0 do 30 tysięcy. Ilość liczb wynosi od 1 tysiąca do 10 tysięcy z krokiem tysiąc. W programie zaimplementowano wstawianie liczb do drzew, wyszukiwanie liczb oraz dla drzewa BST usuwanie liczb. Na wykresach przedstawiono pomierzony czas wstawiania liczb, szukania i usuwania liczb (w przypadku BST) dla 'n' liczb. Program był pisany w wersji python 3.9.13, a wykorzystane biblioteki to: time, random, networkx, matplotlib.pyplot, pygraphviz.

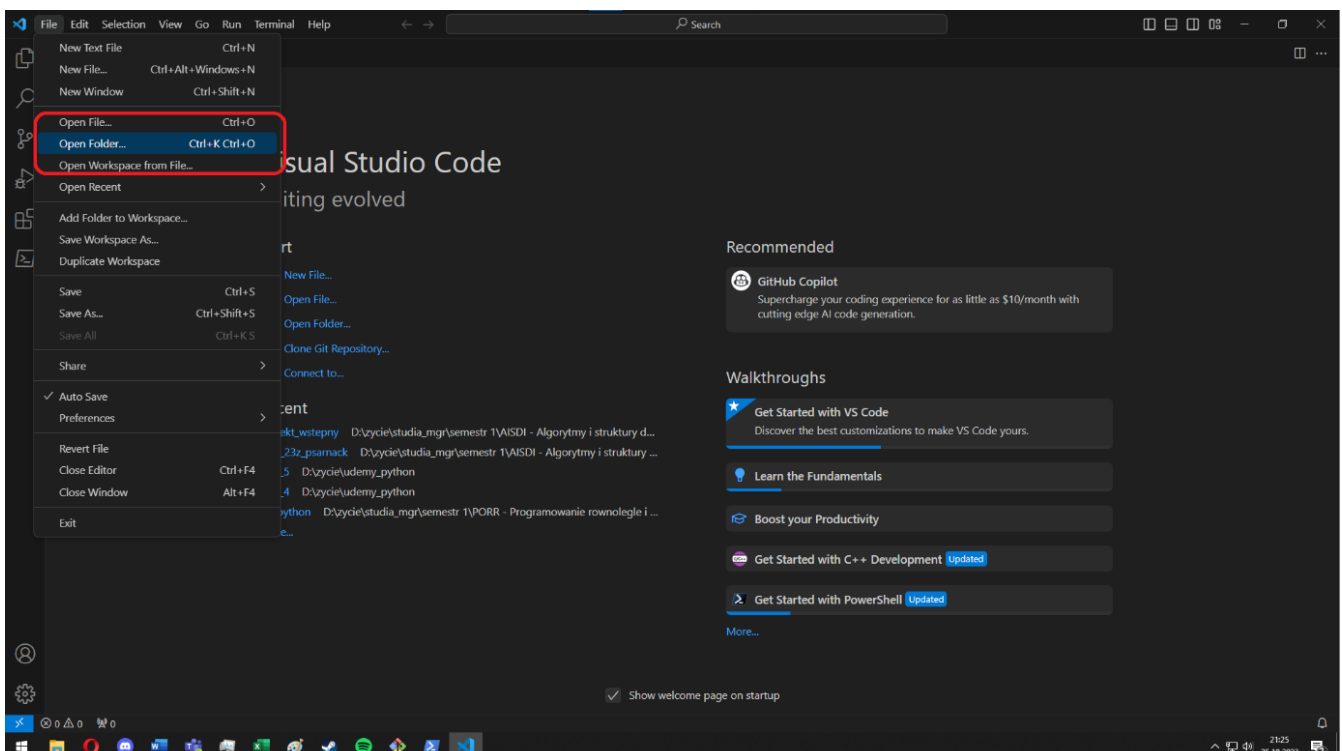
2. Struktura projektu

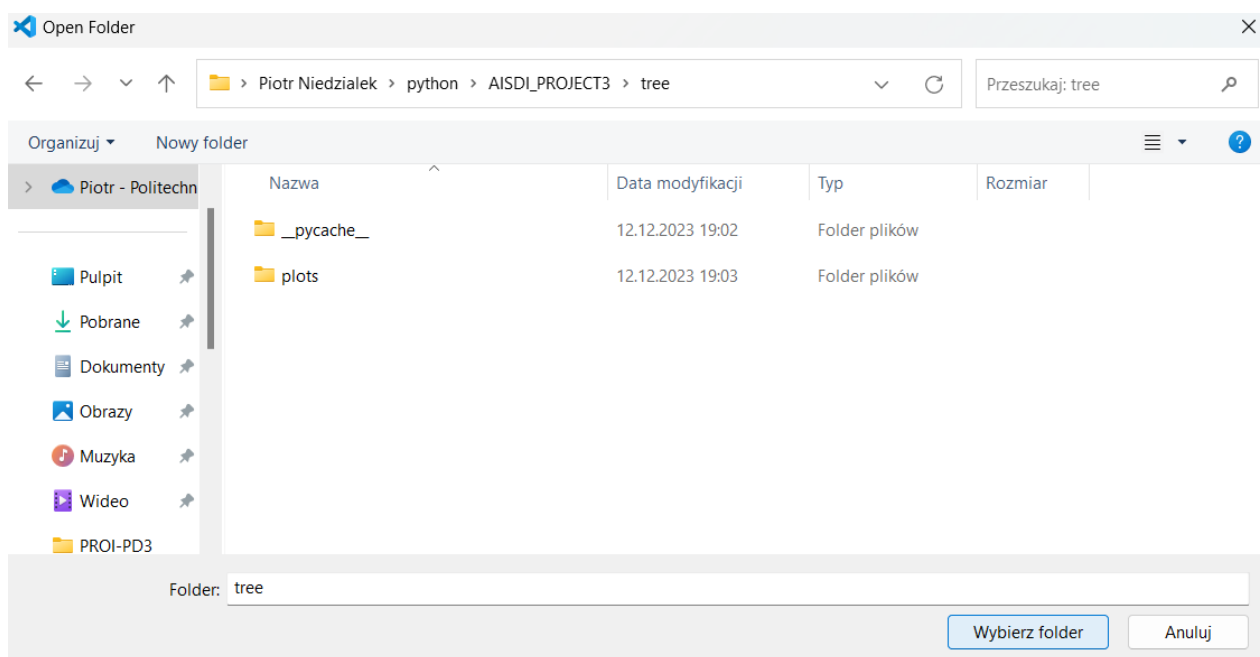
Projekt składa się z:

- Pliku main.py, który zawiera główny program wywołujący funkcje z plików i mierzący czas (Paweł Sarnacki, Piotr Niedziałek)
- Pliku BST.py, który zawiera implementację drzewa BST – klasę BST i metody służące do wstawiania liczb, szukania i usuwania (Piotr Niedziałek)
- Pliku draw_plots.py, który rysuje wykresy porównujące czas działania dla różnych rodzajów kopców i różnej ilości operacji wstawiania i usuwania szczytu (Paweł Sarnacki, Piotr Niedziałek)
- Pliku draw_graphs.py, który rysuje reprezentacje drzew (Paweł Sarnacki)
- Pliku test.py w którym przeprowadzone są testy jednostkowe (Paweł Sarnacki, Piotr Niedziałek)
- Katalogu plots, który zawiera wygenerowane wykresy i grafy

3. Uruchomienie projektu z środowiska Visual Studio Code

3.1. Otworzenie folderu, wybór ścieżki i kliknięcie wybierz folder





3.2. Wybranie terminalu bash, wpisanie „py main.py”, lub kliknięcie F5, wyświetlone i zapisane zostaną wykresy i grafy

4. Uruchomienie projektu bez środowiska Visual Studio Code

Przejdźcie do odpowiedniego folderu z plikami projektu i wpisanie w cmd „py main.py”

```
Wiersz polecenia
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\piotr>cd python

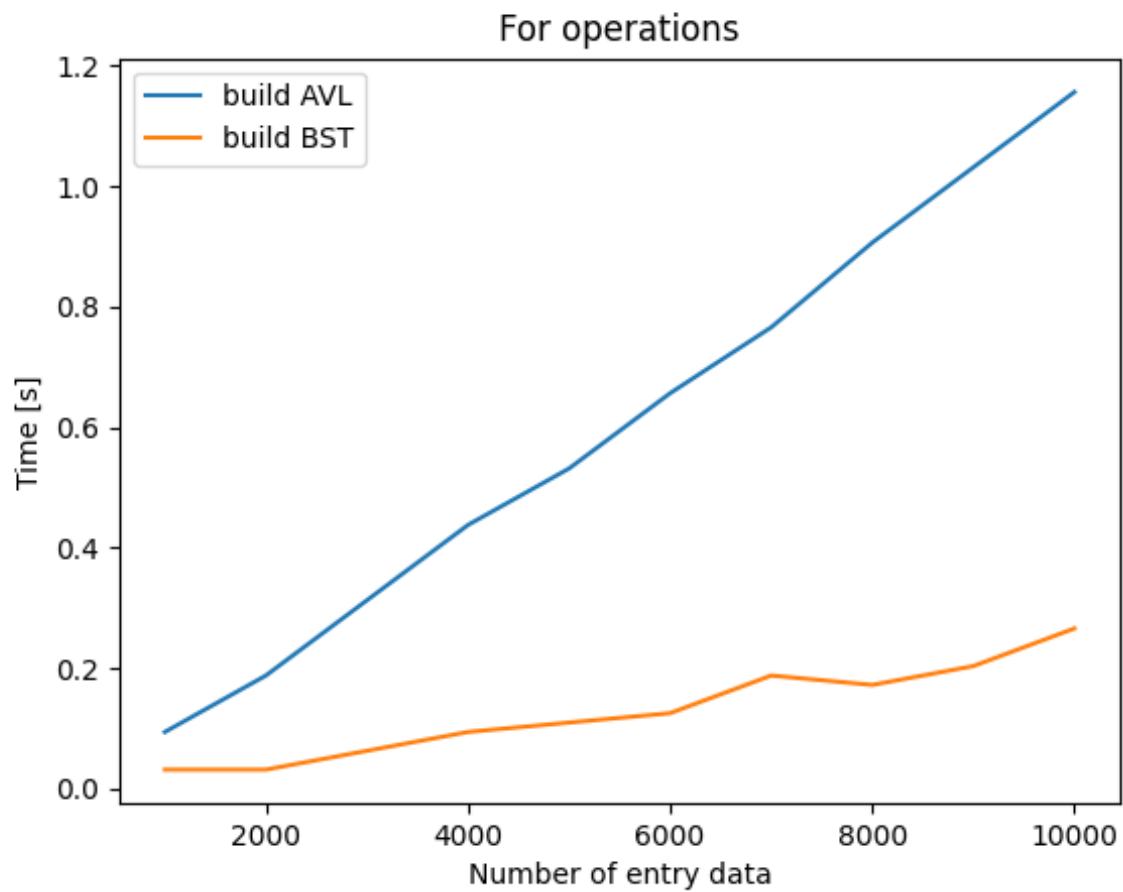
C:\Users\piotr\python>cd AISDI_PROJECT3

C:\Users\piotr\python\AISDI_PROJECT3>cd tree

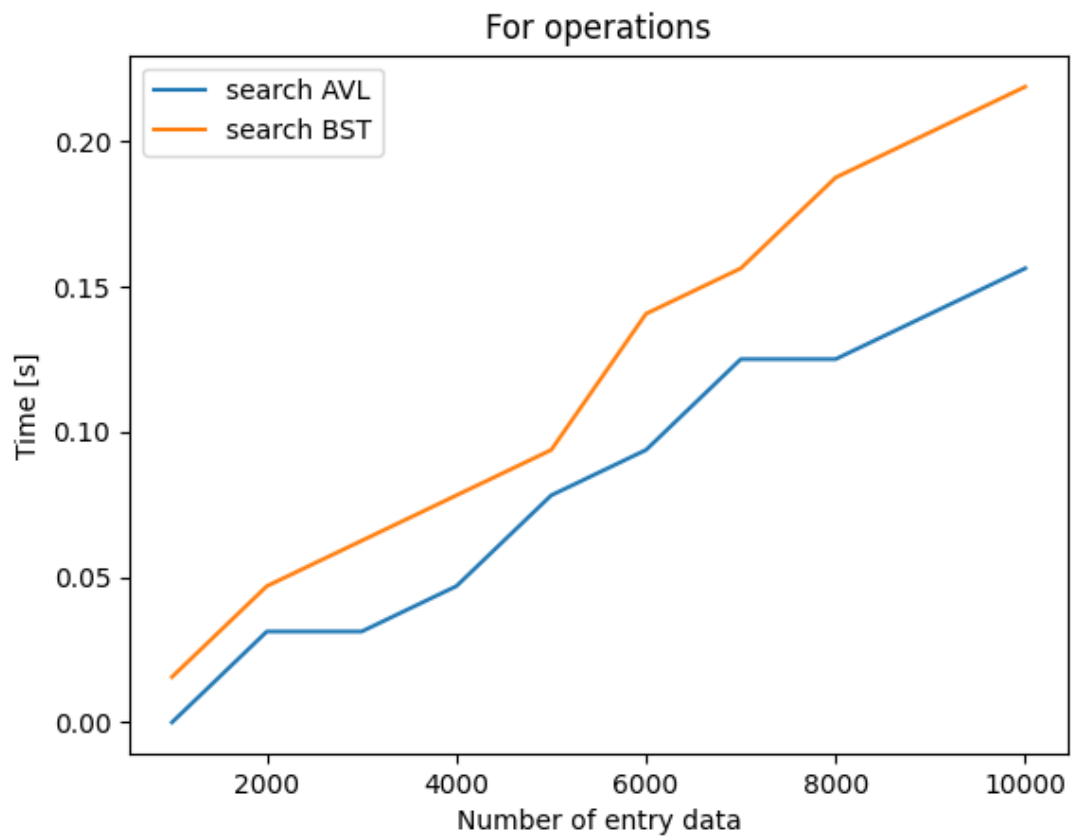
C:\Users\piotr\python\AISDI_PROJECT3\tree>py main.py
```

5. Otrzymane wyniki

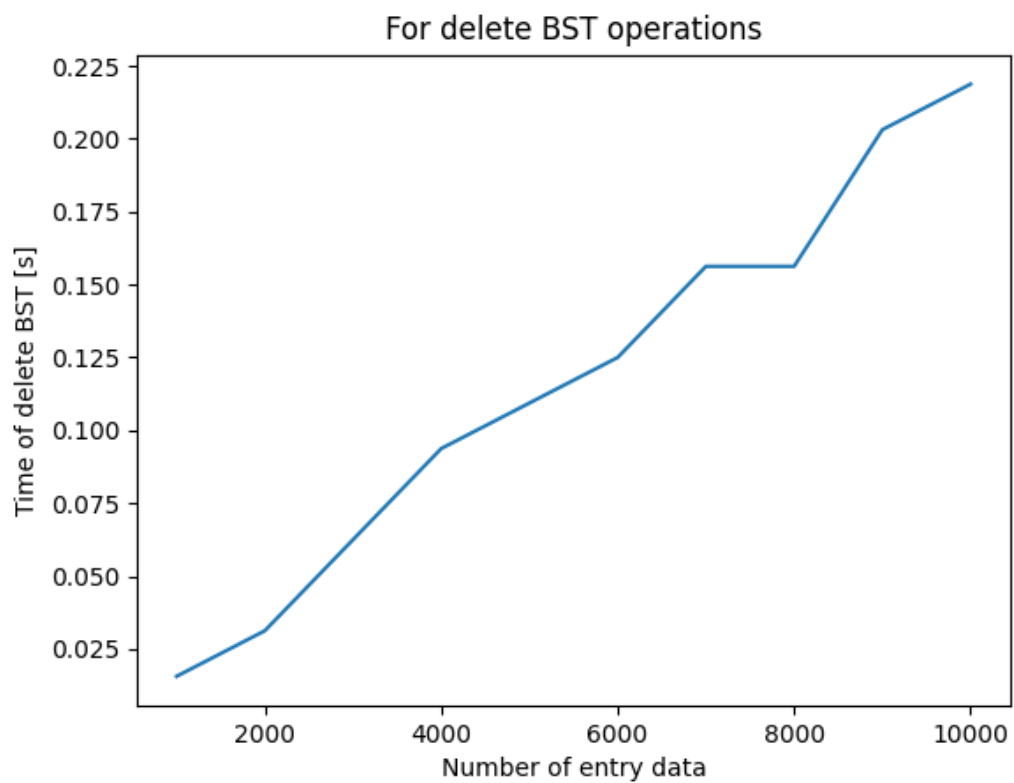
Czas budowania drzew BST i AVL dla kolejnych wartości 'n', czyli ilości elementów:



Czas szukania liczb w drzewach:

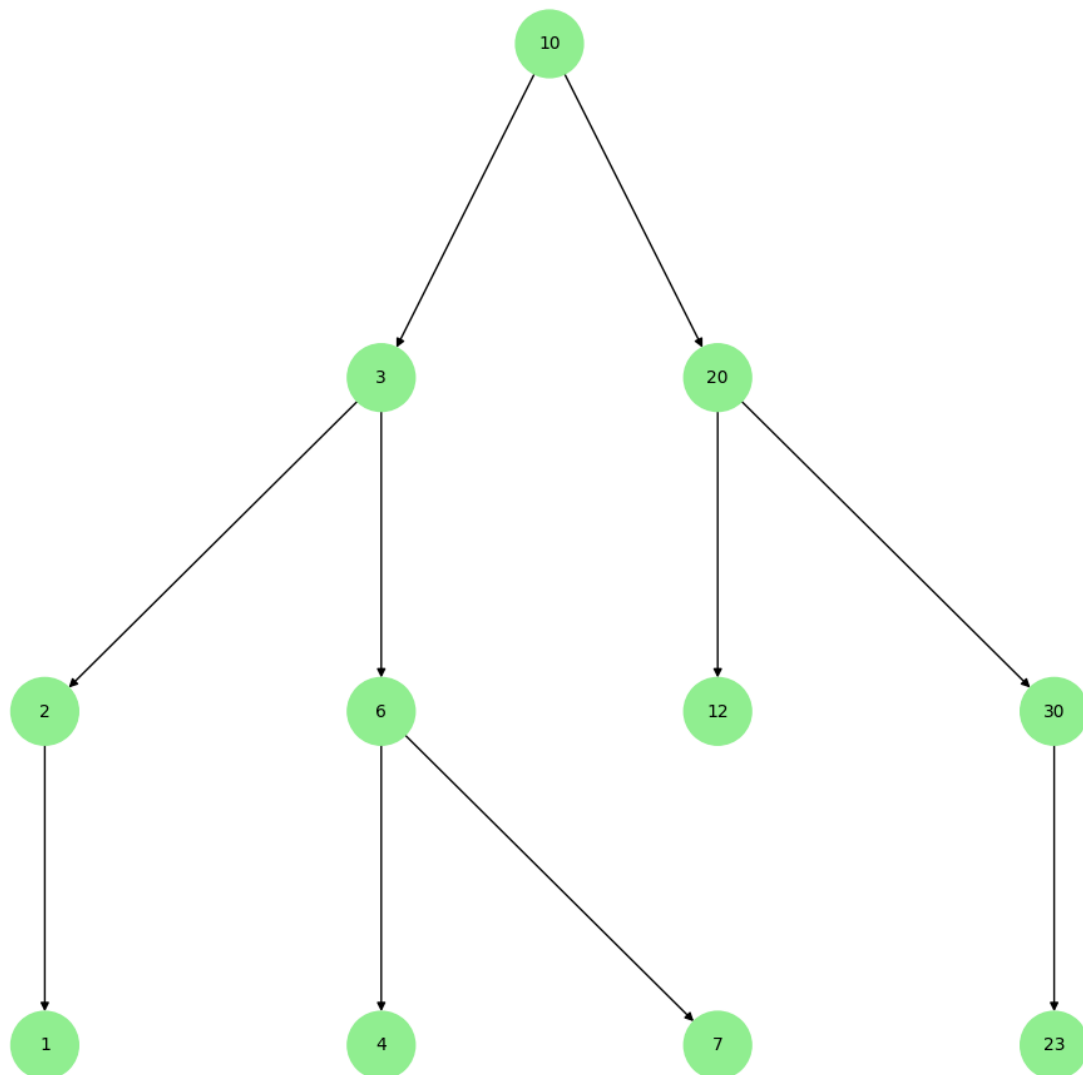


Czas usuwania liczb z drzewa BST:

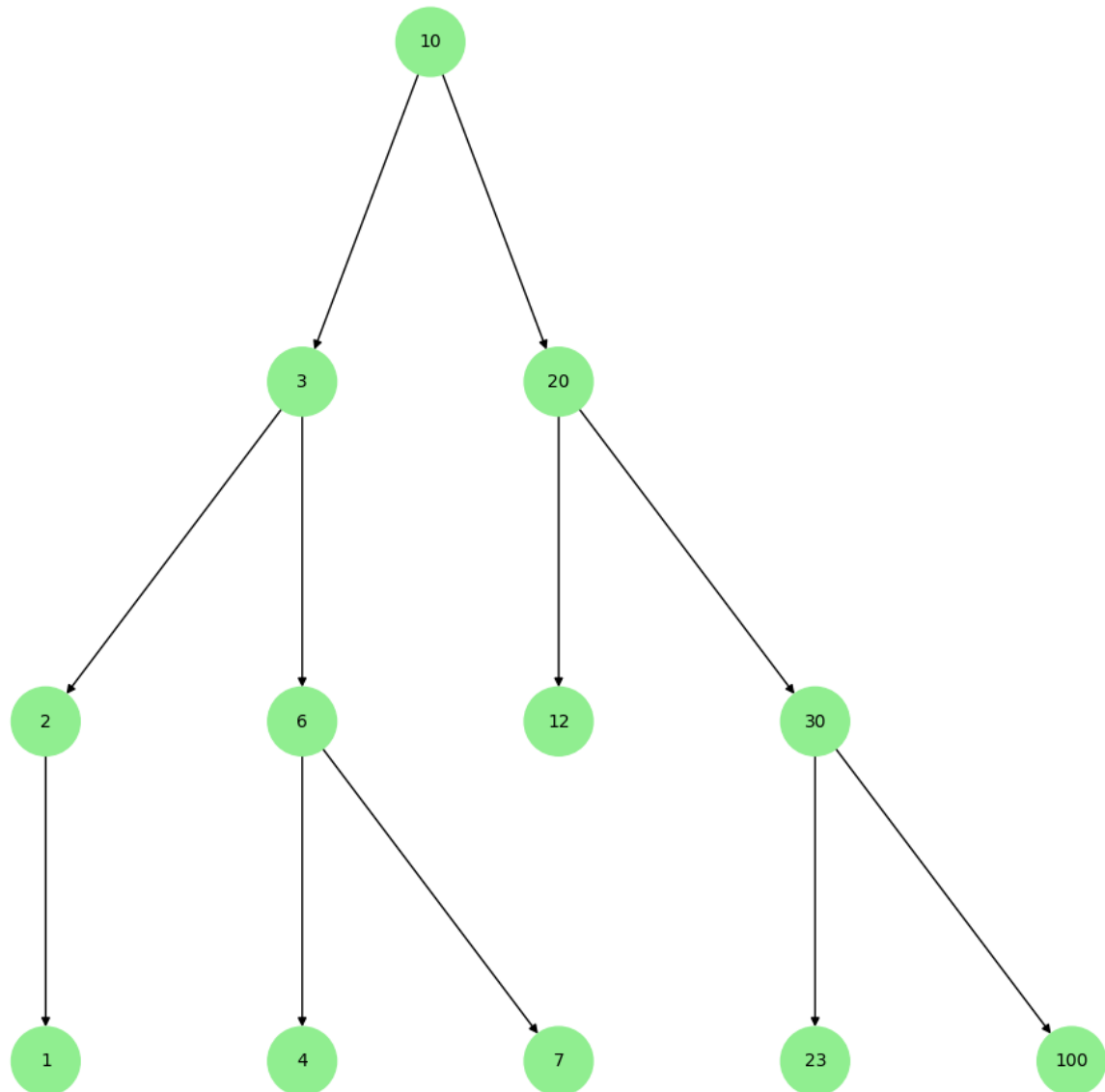


Przykłady wyświetlania drzew:

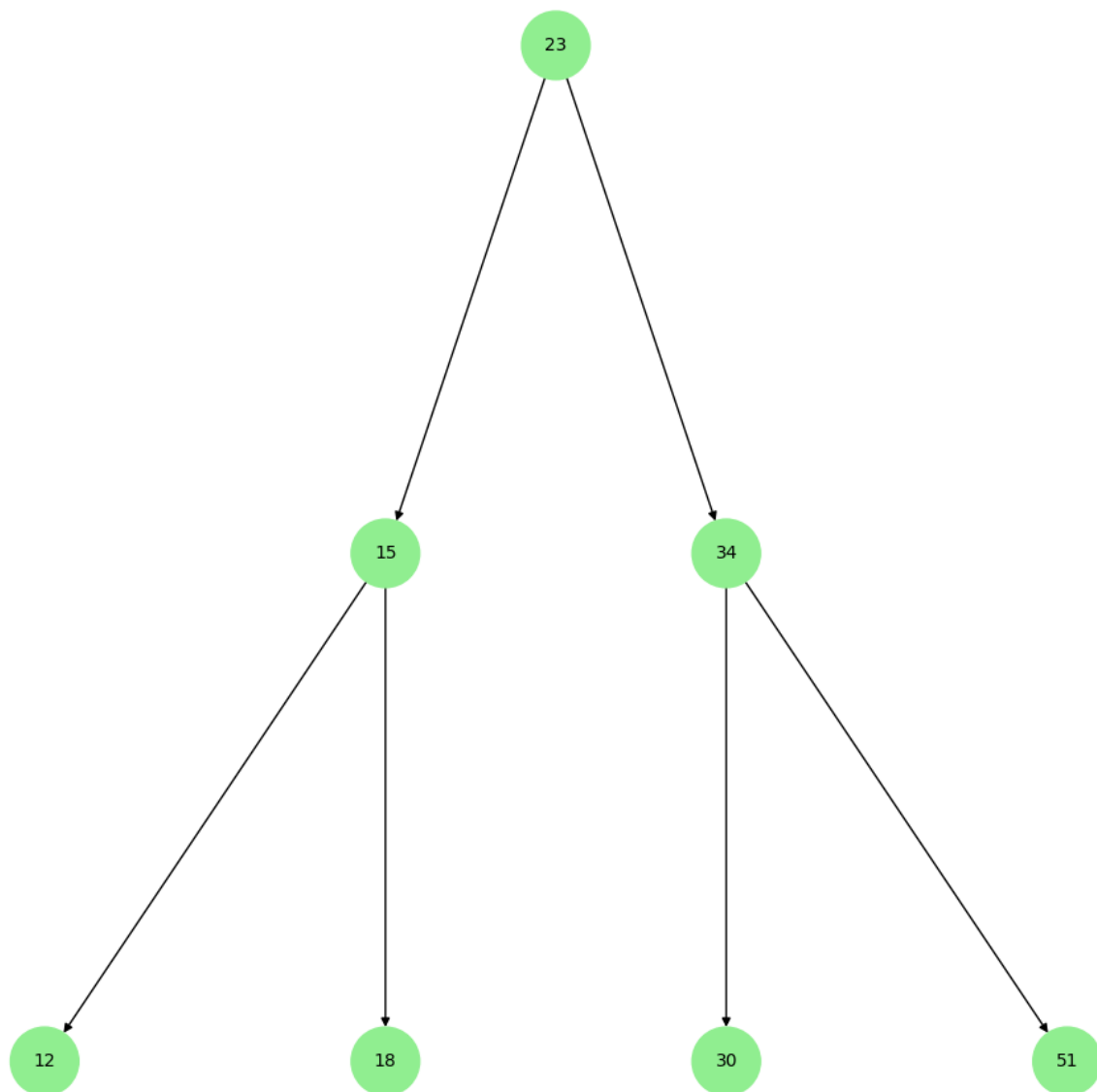
- drzewo AVL



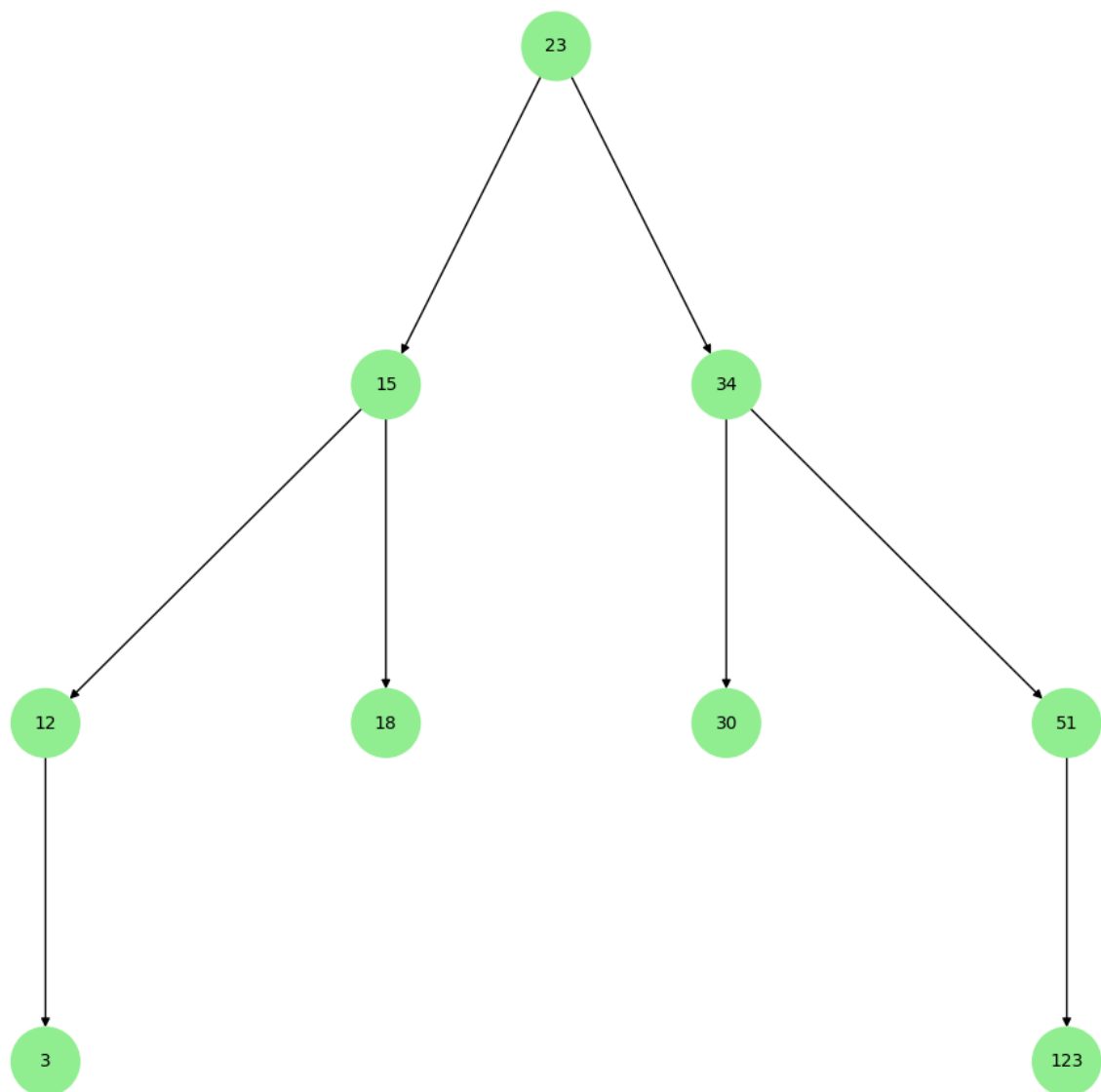
- drzewo AVL po dodaniu liczb



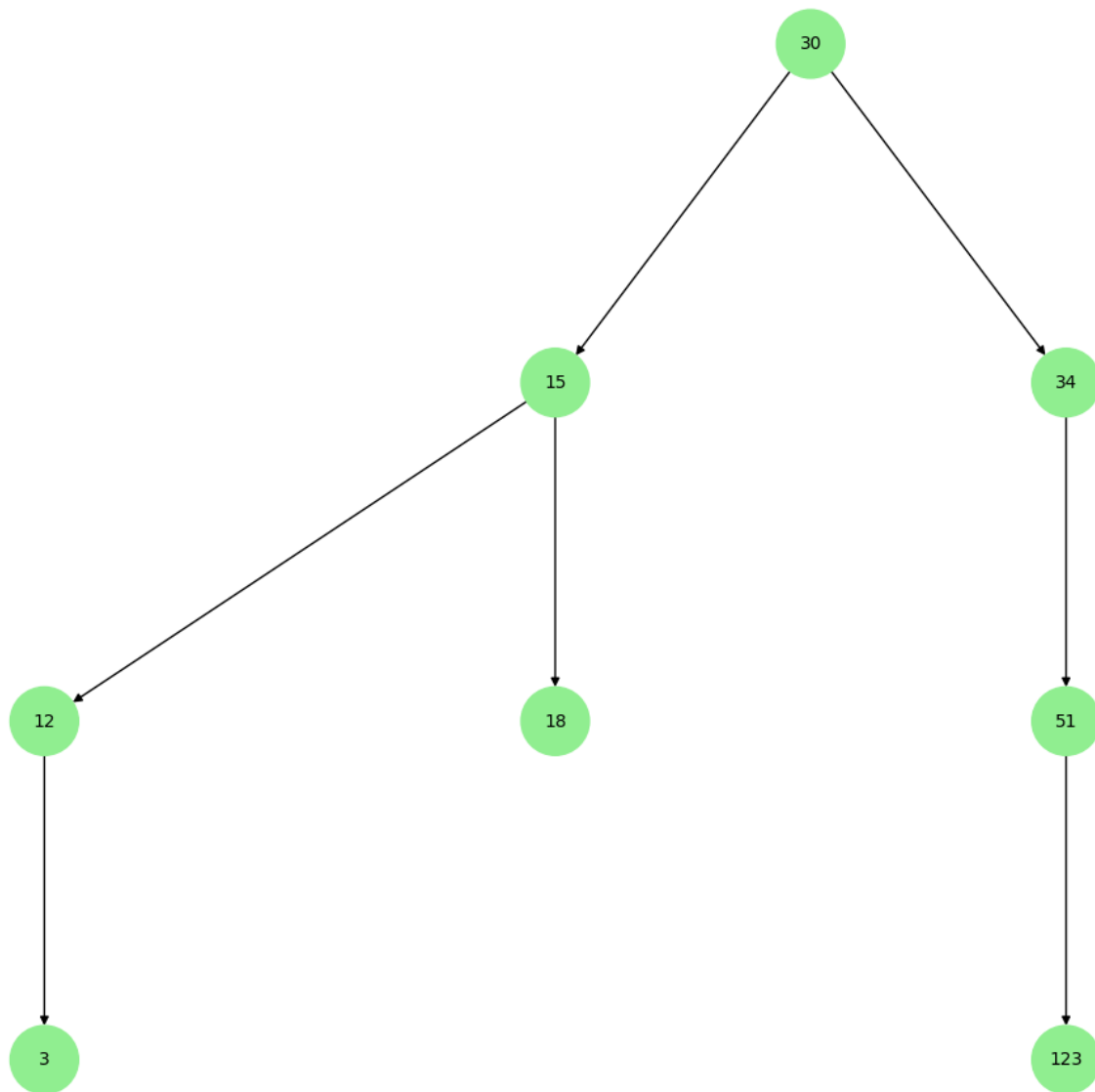
- drzewo BST



- drzewo BST po dodaniu liczb



- drzewo BST po usunięciu liczb:



6. Wnioski

- przedstawione grafy drzew pokazują, że zostały one zaimplementowane poprawnie
- drzewo BST jest budowane prędkiej – ma to sens ponieważ przy budowie drzewa AVL dokonują się rotacje, które mają za zadanie zbalansowanie wysokości drzewa
- wyszukiwanie odbywa się prędkiej dla drzewa AVL – jest to zaleta zbalansowania drzewa
- wykonane testy potwierdzają poprawność działania dodawania, szukania oraz usuwania