

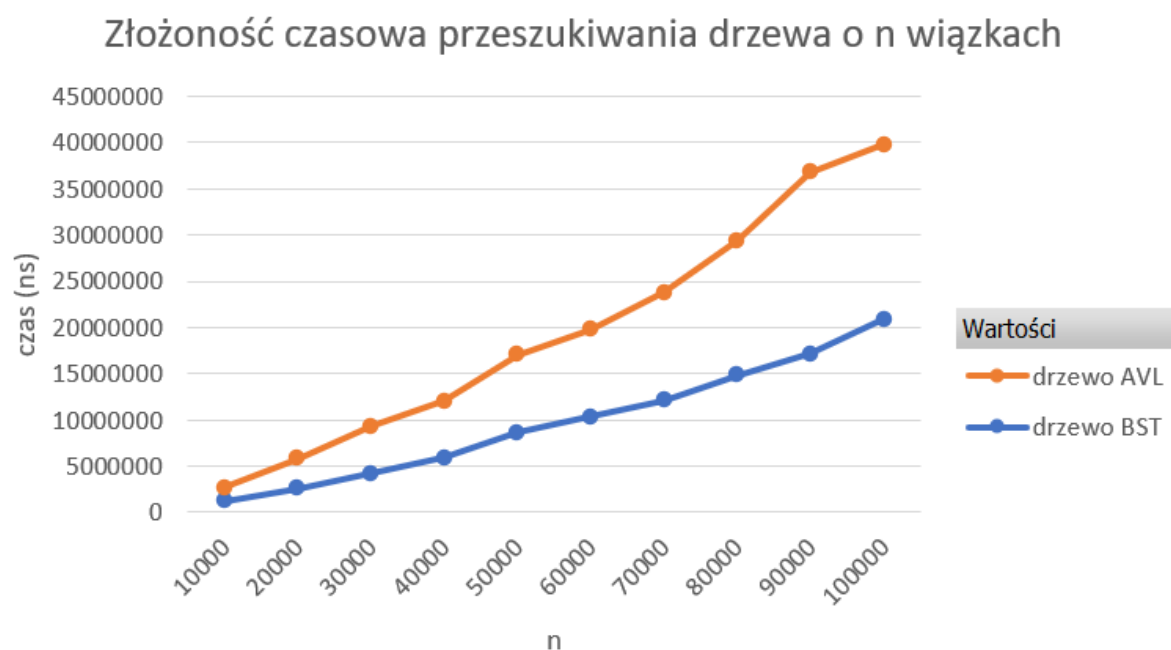
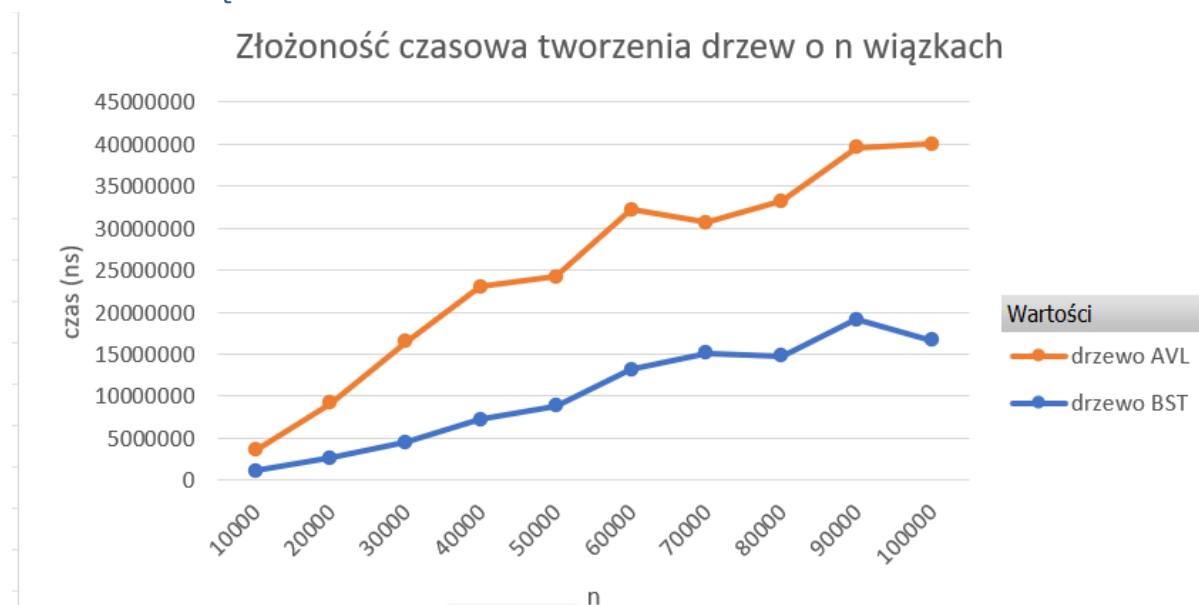
Sprawozdanie- AISDI zadanie 2- M. Bieńkowski, M. Garbowski

Podział ról:

Maksym Bieńkowski - implementacja drzewa BST, przygotowanie sprawozdania

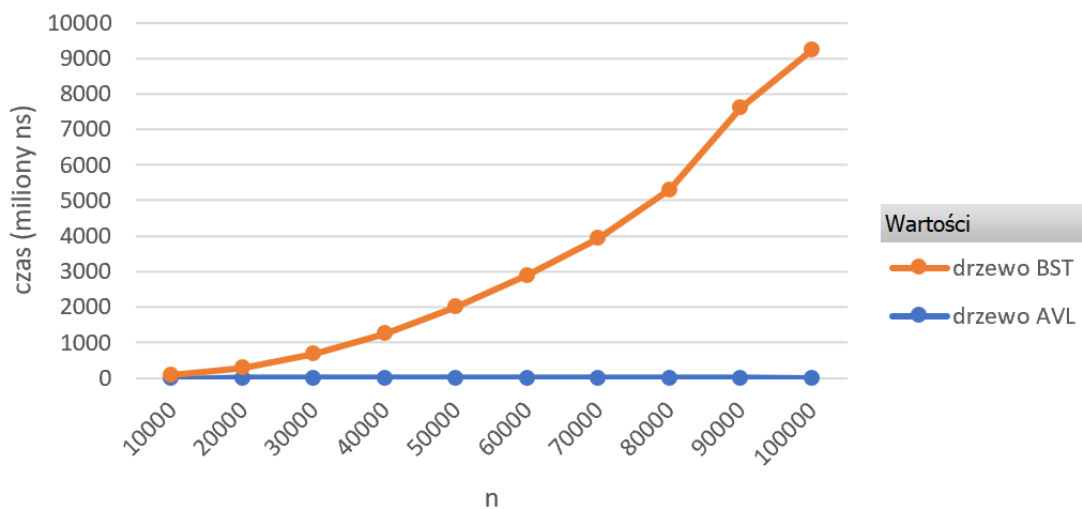
Mikołaj Garbowski - implementacja drzewa AVL, refaktoryzacja kodu drzewa BST

Wykresy obrazujące złożoność czasową wstawiania, szukania i usuwania węzłów w drzewach



drzewo AVL drzewo BST

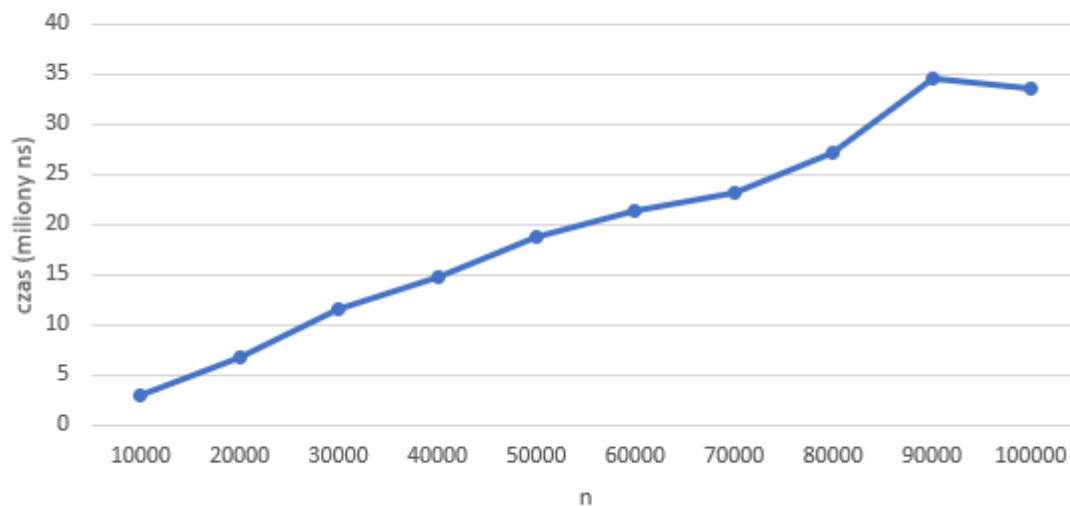
Przeszukiwanie posortowanego drzewa o n węzłach



Rozmiar ▾

Suma z czas usuwania

Czas usuwania z drzewa BST o n węzłach



Rozmiar ▾

Wnioski:

W przeprowadzonych losowych testach oba drzewa wykazały się podobną złożonością czasową - ich wykresy mają jednakowe kształty. Mimo, że dla losowych danych drzewo AVL okazało się wolniejsze, niż BST, należy pamiętać, że jest ono bardziej niezawodne - jego wyższość ukazuje trzeci wykres, przypadek szukania w drzewie stworzonym z posortowanej listy (w przypadku BST powstaje struktura na kształt linked listy, której przeszukiwanie ma złożoność n). Na podstawie kształtu wykresu usuwania węzłów BST stwierdzamy, że złożoność czasowa w średniej sytuacji wynosi $n \log(n)$.

Struktura drzew

BST

```
[60, 60]
  L: [20, 20]
    L: [10, 10]
    R: [30, 30]
      L: [25, 25]
      R: [40, 40]
  R: [70, 70]
    R: [80, 80]
      R: [90, 90]
        R: [100, 100]
```

AVL

```
[60, 60]
  L: [20, 20]
    L: [10, 10]
    R: [30, 30]
      L: [25, 25]
      R: [40, 40]
  R: [80, 80]
    L: [70, 70]
    R: [90, 90]
      R: [100, 100]
```

Oba drzewa zostały wygenerowane za pomocą tej samej sekwencji operacji wstawiania. Widoczna jest różnica w ich strukturze - w przeciwieństwie do BST, drzewo AVL dokonuje automatycznego balansowania przy każdym wstawieniu, co skutkuje niższą wysokością i efektywniejszym wyszukiwaniem.