

**Міністерство освіти і науки України**  
**Львівський національний університет імені Івана Франка**  
**Факультет прикладної математики та інформатики**

Кафедра програмування

Лабораторна робота №11  
**АВЛ-ДЕРЕВА**  
з курсу “Алгоритми та структури даних”

Виконав:  
Студент групи ПМІ-12  
Бенько Володимир Сергійович

Львів – 2023

## АВЛ-дерево

АВЛ-дерево — збалансоване по висоті двійкове дерево пошуку: для кожної його вершини висота її двох піддерев відрізняється не більше ніж на одиницю. АВЛ — абревіатура, утворена першими літерами творців (радянських учених) Адельсон-Вельського Георгія Максимовича і Ландіс Євгена Михайловича. АВЛ-дерево ніколи не буде більш ніж на 45% вище від відповідного ідеального збалансованого дерева, незалежно від кількості вузлів, що дозволяє виконувати більшість операцій над деревом за  $O(\log_2 n)$ .

Клас AVLTree реалізує структуру даних таблиця на АВЛ-дереві. Таблиця дозволяє напряду доступатись до елементів, використовуючи відповідні їм ключі. Клас AVLTree містить приховану структуру Node, яка містить ключ, значення та посилання на корені лівого та правого піддерев заданої вершини та їх висоти, а також методи обходу дерева для очищення, виведення та обчислення висоти піддерев. Також клас AVLTree містить лічильник кількості вершин, конструктор, деструктор та ряд публічних методів:

- Add(k, v) - додає нову вершину в дерево з ключем "k" та значенням "v".
- Find(k) - знаходить вершину з ключем "k" і повертає вказівник на її значення.
- Replace(k, v) - замінює значення вершини з ключем "k" на нове значення "v".
- Remove(k) - видаляє вершину з дерева з ключем "k".
- Clear() – видаляє всі вершини дерева.
- GetCount() – повертає кількість вершин дерева.
- GetHeight() – повертає висоту дерева.
- Output(ostream out) – виводить дерево в потік out.

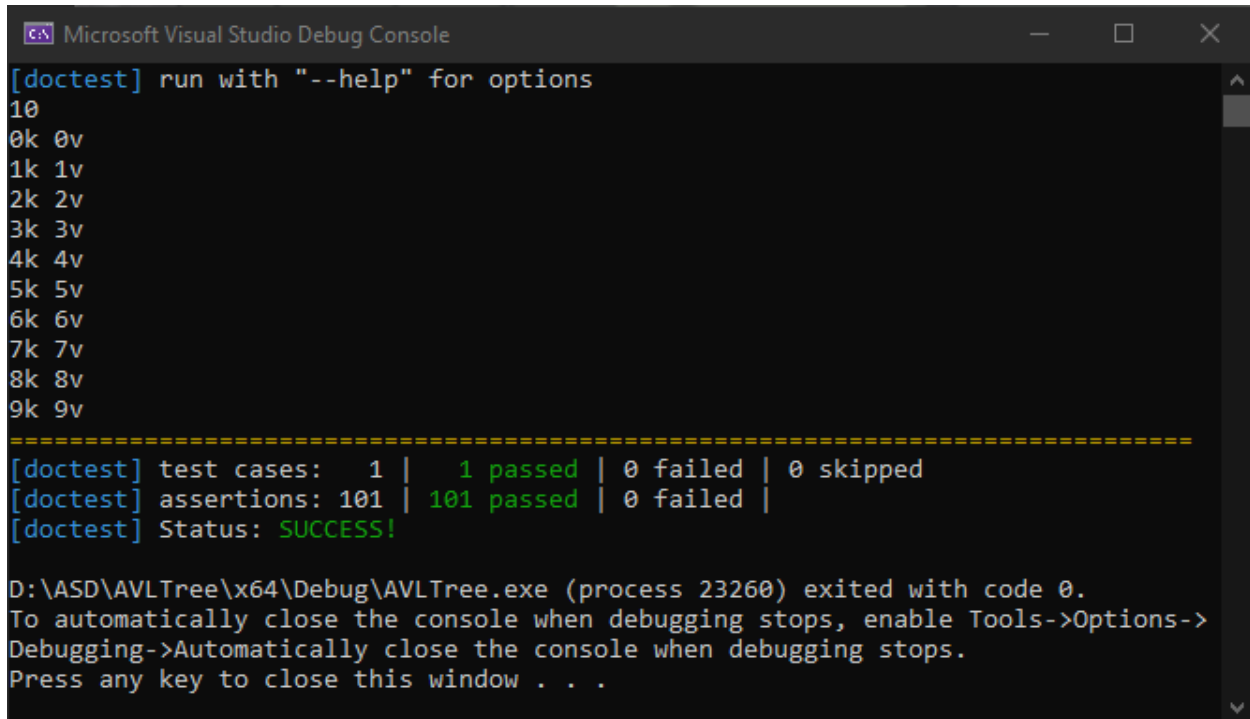
### Складність роботи методів:

- Add :  $O(\log_2 n)$
- Find :  $O(\log_2 n)$
- Replace :  $O(\log_2 n)$
- Remove :  $O(\log_2 n)$
- Clear :  $O(n)$
- Output :  $O(n)$
- GetHeight :  $O(n)$
- GetCount :  $O(1)$

**Просторова складність АВЛ-дерева:**  $O(n)$

### Приклад:

Щоб переконатись, що всі методи класу AVLTree працюють правильно, в програмі написані юніт-тести. Усі вони проходять успішно:



```
Microsoft Visual Studio Debug Console

[doctest] run with "--help" for options
10
0k 0v
1k 1v
2k 2v
3k 3v
4k 4v
5k 5v
6k 6v
7k 7v
8k 8v
9k 9v

=====
[doctest] test cases: 1 | 1 passed | 0 failed | 0 skipped
[doctest] assertions: 101 | 101 passed | 0 failed |
[doctest] Status: SUCCESS!

D:\ASD\AVLTree\x64\Debug\AVLTree.exe (process 23260) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

### Висновок:

AVL-дерева гарантують збалансованість тому вони є ефективними для пошуку, вставки та видалення елементів. Однак, повороти можуть бути дорогими в операціях, тому AVL-дерева можуть бути менш ефективними в порівнянні з іншими структурами даних, такими як хеш-таблиці, якщо часто відбувається зміна даних у дереві.