Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №3

“Розширюване дерево”

Виконав студент 2-го курсу

Групи К-29

Лавров Сергій Сергійович

2021

**Предметна область**

Футбол

Об'єкти: Команди, Гравці

Примітка: Маємо команди. Для кожної команди визначені гравці

**Завдання**

Реалізуйте розширюване дерево (splay tree).

**Теорія**

Розширюване дерево – це двійкове дерево пошуку з підтримкою збалансованості.

• Не потребує додаткових полів у вузлі.

• Явні функції балансування відсутні.

• При кожному звертанні до дерева виконується

«операція розширення» (splay operation).

• В результаті вузли, до яких звертаються частіше,

зберігаються ближче до кореня, а до яких рідше –

ближче до листків.

**Алгоритми**

1)Splay. Функція для переміщення вершини в корінь за допомогоюоперацій: Zig, Zig-Zig і Zig-Zag.

2)Search. Спочатку звичайний пошук. При знаходженні елементу запускаємо Splay для нього.

3)Insert. Запускаємо Split від елементу, що додається, і підвішуємо дерева, що вийшли, за нього.

4)Delete. Знаходимо елемент в дереві, робимо Splay для нього, робимо поточним деревом Merge його дітей.

**Складність**

Всі операції виконуються в середньому за O(log n)

**Мова програмування**

С++

**Модулі програми**

class Team, class Player – класи для реалізації предметної області.

Для класу Team перевизначив оператори <, >, !=, == для роботи з розшиюваним деревом, і оператор << для простішого виводу в консоль.

class SplayTree клас розширюваного дерева

size\_t size кількість елементів в дереві

Node<T>\* root корінь дерева

**Основні функції**

void rotateLeft(Node<T>\*& theRoot, Node<T>\* x) функція повороту вліво

void rotateRight(Node<T>\*& theRoot, Node<T>\* x) функція повороту вправо

void splay(Node<T>\*& theRoot, Node<T>\* theNode) функція реалізовує алгоритм Splay

void insert(T element) Вставка елемента в дерево

void erase(T element) Видалення елемента з дерева

T\* search(T element) Функція пошуку елемента в дереві

**Допоміжні функції**

void freeMemory(Node<T>\* theNode) Функція для звільненя динамічної пам'яті

Node<T>\* getMaxNode(Node<T>\* theNode) Функція для отримання максимальної вершини

void printUtil(Node<T>\* root, int space) Допоміжна функція для виводу дерева в консоль

void print() Функція для виводу дерева в консоль

**Інтерфейс користувача**

Користувач може вводити наступні команди в консоль:

addteam ‘назва команди’ ‘кількість гравців’ ‘імена’…

Створює нову команду і додає її до розширюваного дерева

insertplayers ‘назва команди’ ‘кількість гравців’ ‘імена’…

Шукає команду за назвою і вставляє в неї нових гравців

search ‘назва команди’

Знаходить команду в дереві і виводить її гравців в консоль

erase ‘назва команди’

Видаляє команду з дерева за назвою

print

Виводить дерево в консоль

exit

Вихід з програми

**Тестовий приклади**

Додаємо 5 команд

addteam Ukraine 3 Schevchenko Konoplianko Iarmolenko

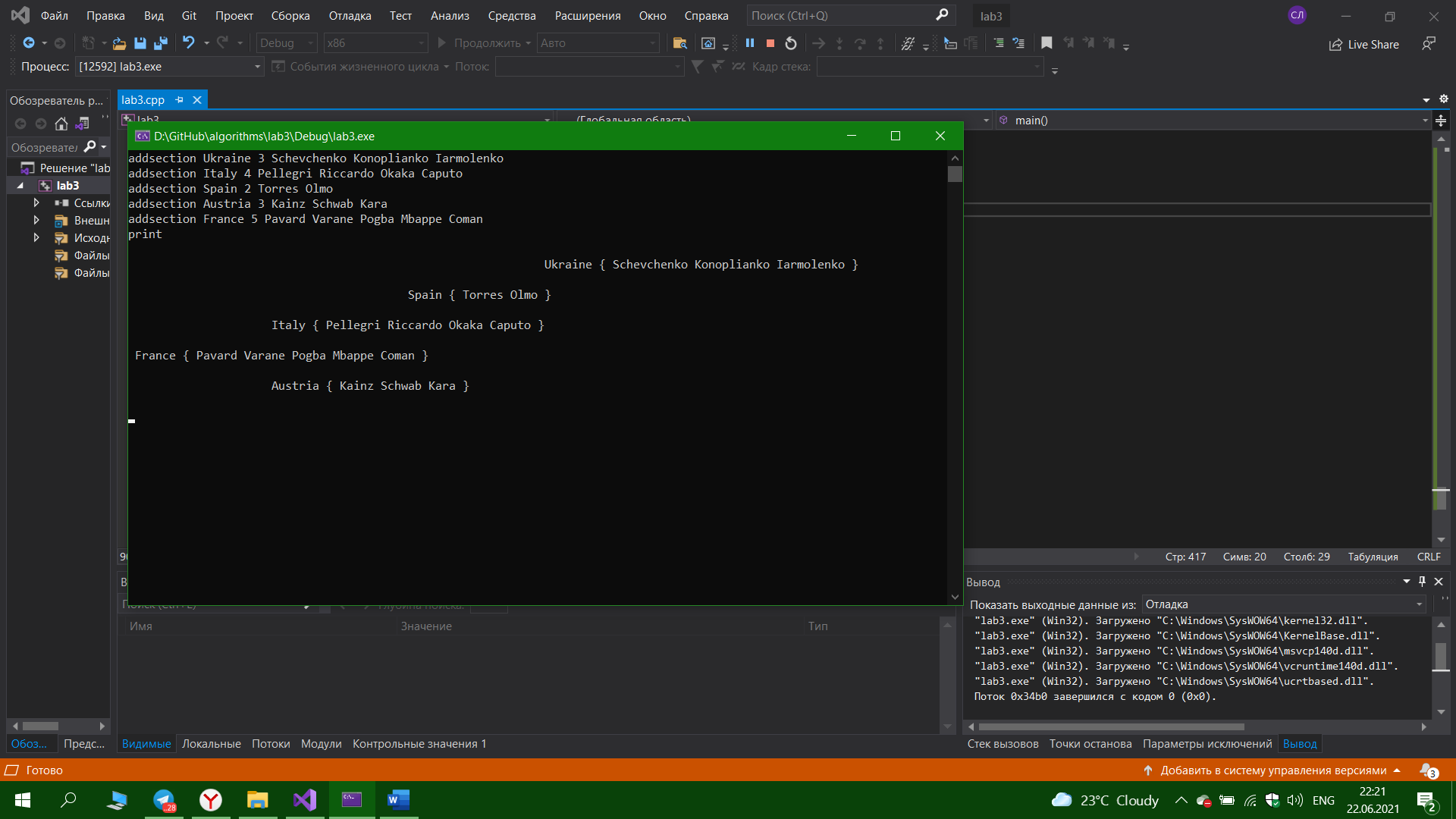
addteam Italy 4 Pellegri Riccardo Okaka Caputo

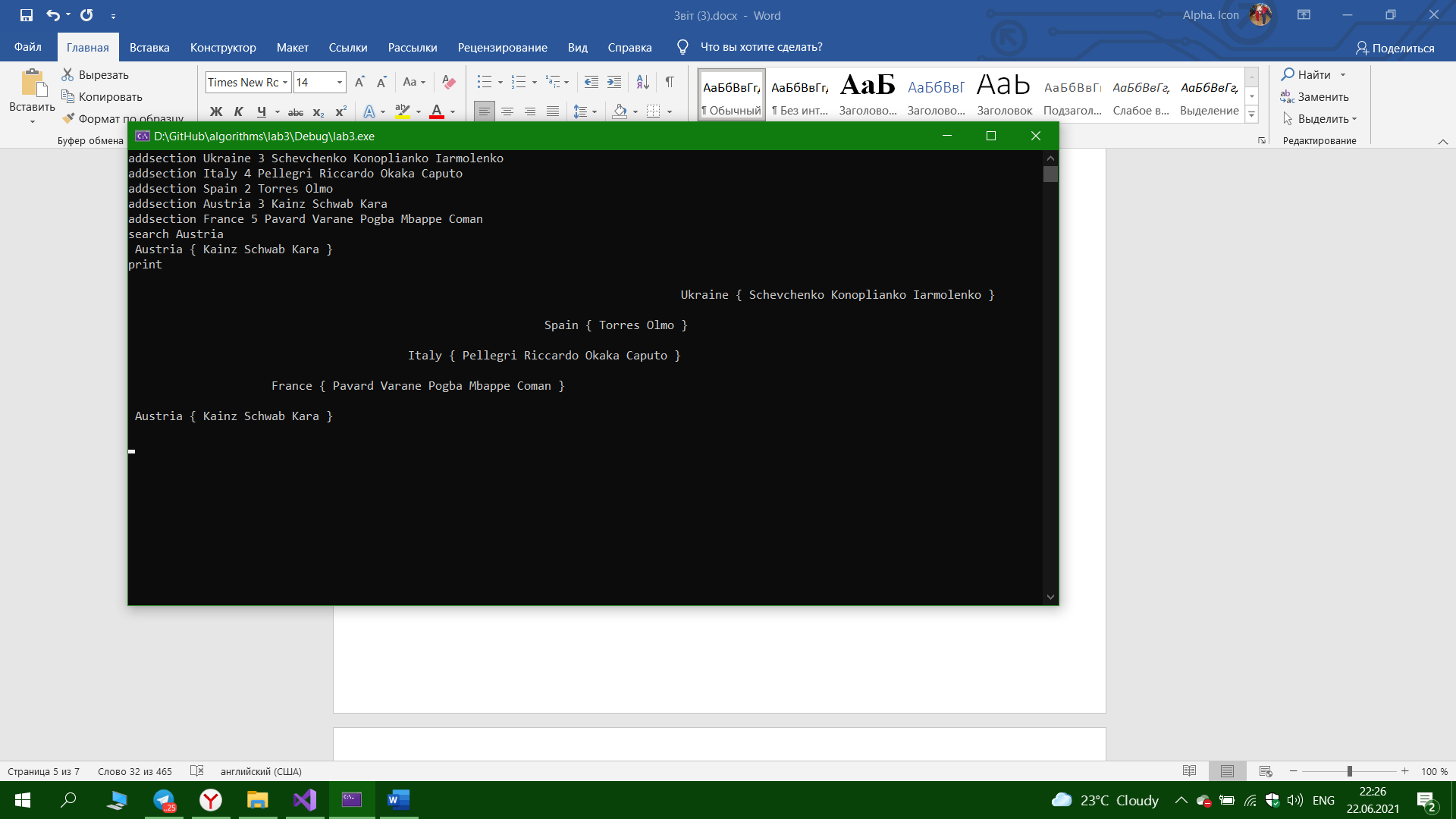
addteam Spain 2 Torres Olmo

addteam Austria 3 Kainz Schwab Kara

addteam France 5 Pavard Varane Pogba Mbappe Coman

Виведемо дерево

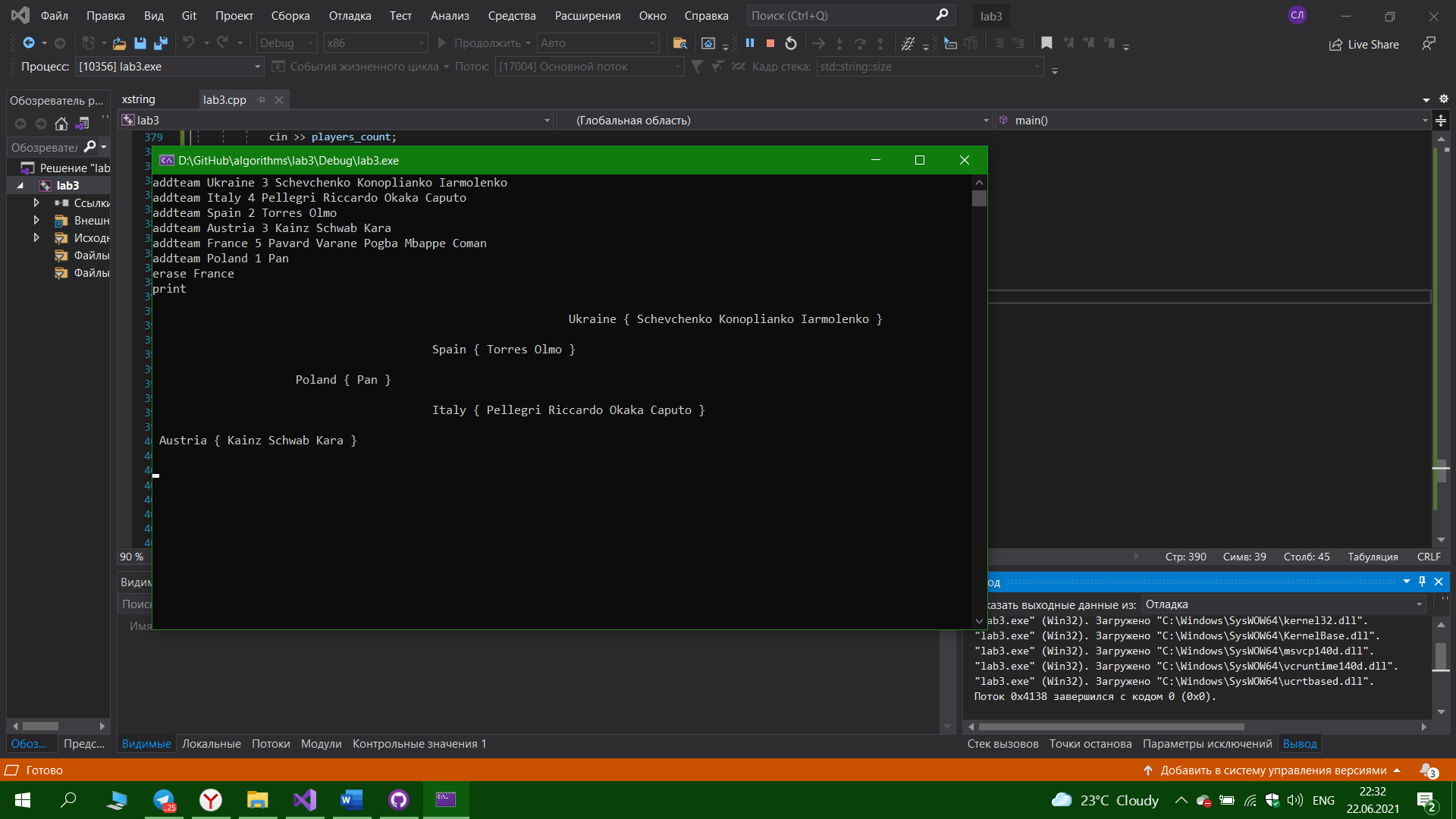
Знайдемо Austria і виведемо дерево



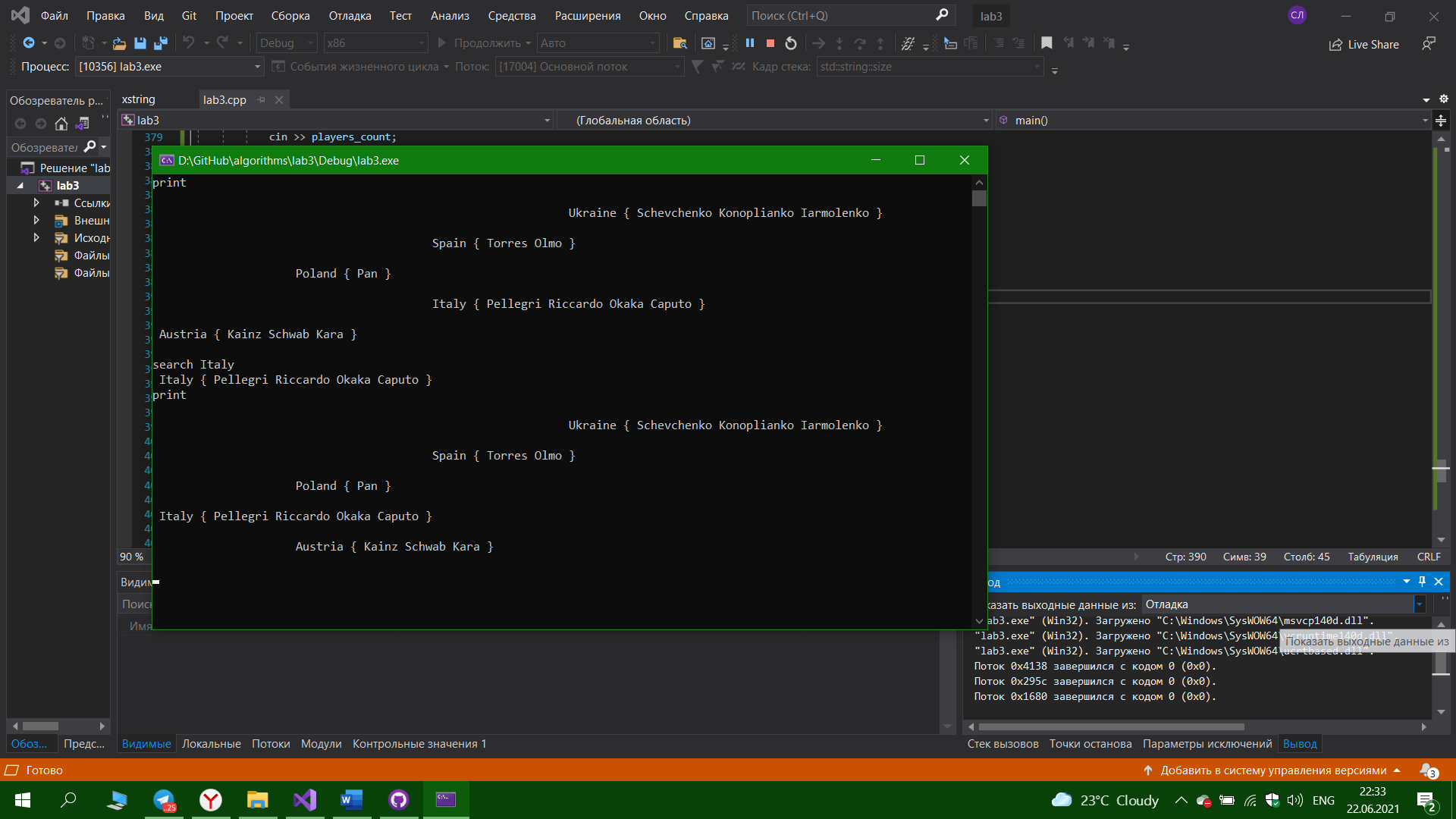
Вставимо і видалимо одну команду. Виведемо дерево

addteam Poland 1 Pan

erase France



Знайдемо Italy і виведемо дерево



**Висновки**

Розширюване дерево – це бінарне дерево пошуку з підтримкою збалансованості, в якому вершини з частішим зверненням зберігаються ближче до кореня і не потребують додаткових змінних.

**Література**

* Лекція №3
* neerc.ifmo.ru/wiki/index.php?title=Splay-дерево