Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

A blue and white logo

Description automatically generated

**Звіт**

**Звіт**

**про виконання лабораторних та практичних робіт блоку № 6**

На тему: «Програмування: алгоритм, програма, код. Системи числення. Двійкова система числення. Розробка та середовище розробки програми.»

***з дисципліни:*** «Основи програмування»

до:

Практичних Робіт до блоку № 6

**Виконав:**

Студент групи ШІ-11

Корнілов Артем Сергійович

Львів 2024

**Тема:** Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.

**Мета:** Засвоїти основи роботи з динамічними структурами даних, такими як черга, стек, списки та дерева. Ознайомитися з алгоритмами їх обробки для розв'язання різноманітних задач.

**Теоретичні відомості та джерела:**

**Динамічні структури**

**-**[Урок №89. Динамічне виділення пам’яті;](https://acode.com.ua/urok-89-dynamichne-vydilennya-pam-yati/)

**-**[Урок №90. Динамічні масиви;](file:///C:\Users\User\Downloads\Урок%20№90.%20Динамічні%20масиви)

-Стек;

[C++ ⦁ Теорія ⦁ Урок 58 ⦁ Стек, Куча, Статична пам'ять](https://www.youtube.com/watch?v=NyOjKd5Qruk)

<https://www.youtube.com/watch?v=B3VHHfMW0Pg>

-Черга;

[#4](https://www.youtube.com/watch?v=B3VHHfMW0Pg)

[#5](https://www.youtube.com/watch?v=fmHyFTji-Lc)

-Списки:

[#1](https://www.youtube.com/watch?v=C9FK1pHLnhI&list=PLQOaTSbfxUtAIipl4136nwb4ISyFk8oI4)

[#2](https://www.youtube.com/watch?v=lQ-lPjbb9Ew&list=PLQOaTSbfxUtAIipl4136nwb4ISyFk8oI4&index=6)

[Урок #133](https://www.youtube.com/watch?v=SajrPhE6FoQ&list=PLQOaTSbfxUtAIipl4136nwb4ISyFk8oI4&index=2)

[Урок #134](https://www.youtube.com/hashtag/134https:/www.youtube.com/watch?v=8E6cPXE-IIA)

[Урок #135](https://www.youtube.com/watch?v=tj17qdFMj54&t=1439s)

-Дерева:

[#3](https://www.youtube.com/watch?v=9o_i0zzxk1s)

[C++ ⦁ Теорія ⦁ Урок 144 ⦁ ADT ⦁ Бінарне дерево](https://www.youtube.com/watch?v=qBFzNW0ALxQ)

**Виконання роботи:**

**VNS Lab 10v2:**

1. Написати функцію для створення списку. Функція може створювати порожній список, а потім додавати в нього елементи.

2. Написати функцію для друку списку. Функція повинна передбачати вивід повідомлення, якщо список порожній.

3. Написати функції для знищення й додавання елементів списку у відповідності зі своїм варіантом.

4. Виконати зміни в списку й друк списку після кожної зміни. 5. Написати функцію для запису списку у файл.

6. Написати функцію для знищення списку.

7. Записати список у файл, знищити його й виконати друк (при друці повинне бути видане повідомлення "Список порожній").

8. Написати функцію для відновлення списку з файлу.

9. Відновити список і роздрукувати його.

10.Знищити список.

Записи в лінійному списку містять ключове поле типу int. Сформувати однонаправлений список. Знищити з нього елемент із заданим ключем, додати елемент перед елементом із заданим ключем;

**Algotester Lab 5v3:**

У вас є карта гори розміром N×MN×M.

Також ви знаєте координати {x,y}{x,y} , у яких знаходиться вершина гори.

Ваше завдання - розмалювати карту таким чином, щоб найнижча точка мала число 0, а пік гори мав найбільше число.

Клітинкі які мають суміжну сторону з вершиною мають висоту на один меншу, суміжні з ними і не розфарбовані мають ще на 1 меншу висоту і так далі.

**Algotester Lab 7-8 v1:**

Ваше завдання - власноруч реалізувати структуру даних "Двозв’язний список".  
Ви отримаєте QQ запитів, кожен запит буде починатися зі слова-ідентифікатора, після якого йдуть його аргументи.  
  
Вам будуть поступати запити такого типу:

* Вставка:  
  Ідентифікатор - insertinsert  
  Ви отримуєте ціле число indexindex елемента, на місце якого робити вставку.  
  Після цього в наступному рядку рядку написане число NN - розмір списку, який треба вставити.  
  У третьому рядку NN цілих чисел - список, який треба вставити на позицію indexindex.
* Видалення:  
  Ідентифікатор - eraseerase  
  Ви отримуєте 2 цілих числа - indexindex, індекс елемента, з якого почати видалення та nn - кількість елементів, яку треба видалити.
* Визначення розміру:  
  Ідентифікатор - sizesize  
  Ви не отримуєте аргументів.  
  Ви виводите кількість елементів у списку.
* Отримання значення ii-го елементу  
  Ідентифікатор - getget  
  Ви отримуєте ціле число - indexindex, індекс елемента.  
  Ви виводите значення елемента за індексом.
* Модифікація значення ii-го елементу  
  Ідентифікатор - setset  
  Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення.
* Вивід списку на екран  
  Ідентифікатор - printprint  
  Ви не отримуєте аргументів.  
  Ви виводите усі елементи списку через пробіл.  
  Реалізувати використовуючи перегрузку оператора <<

**Class Practice Task:**

## Задача №1 - Реверс списку (Reverse list)

***Реалізувати метод реверсу списку:*** Node\* reverse(Node \*head);

*Умови задачі:*

-       використовувати цілочисельні значення в списку;

-       реалізувати метод реверсу;

-       реалізувати допоміжний метод виведення вхідного і обернутого списків;

## Задача №2 - Порівняння списків

bool compare(Node \*h1, Node \*h2);

*Умови задачі:*

-       використовувати цілочисельні значення в списку;

-       реалізувати функцію, яка ітеративно проходиться по обох списках і порівнює дані в кожному вузлі;

-       якщо виявлено невідповідність даних або якщо довжина списків різна (один список закінчується раніше іншого), функція повертає ***false***.

## Задача №3 – Додавання великих чисел

Node\* add(Node \*n1, Node \*n2);

*Умови задачі:*

-       використовувати цифри від 0 до 9 для значень у списку;

-       реалізувати функцію, яка обчислює суму двох чисел, які збережено в списку; молодший розряд числа записано в голові списка (напр. 379  ⟹  9→7→3);

-       функція повертає новий список, передані в функцію списки не модифікуються.

## Задача №4 - Віддзеркалення дерева

TreeNode \*create\_mirror\_flip(TreeNode \*root);

*Умови задачі:*

-       використовувати цілі числа для значень у вузлах дерева

-       реалізувати функцію, що проходить по всіх вузлах дерева і міняє місцями праву і ліву вітки дерева

-       функція повертає нове дерево, передане в функцію дерево не модифікується

## Задача №5 - Записати кожному батьківському вузлу суму підвузлів

void tree\_sum(TreeNode \*root);

*Умови задачі:*

-       використовувати цілочисельні значення у вузлах дерева;

-       реалізувати функцію, яка ітеративно проходить по бінарному дереві і записує у батьківський вузол суму значень підвузлів

-       вузол-листок не змінює значення

-       значення змінюються від листків до кореня дерева

**Self Practice**

**Task** **Algotester Lab 5v2:**

У вас є карта гори розміром N×M.

Також ви знаєте координати {x,y} , у яких знаходиться вершина гори.

Ваше завдання - розмалювати карту таким чином, щоб найнижча точка мала число 0, а пік гори мав найбільше число.

Клітинкі які мають суміжну сторону з вершиною мають висоту на один меншу, суміжні з ними і не розфарбовані мають ще на 1 меншу висоту і так далі.

**Код програми з посиланням на зовнішні ресурси**

[**VNS Lab 10v2**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/vns_lab_10_task_1_variant_2_artem_kornilov.cpp)**:**

**затратність ~3 години**

[**Algotester Lab 5v3**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/algotester_lab_5_variant_3_artem_kornilov.cpp)**:**

**затратність ~ 2 години**

[**Algotester Lab 7-8 v1**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/algotester_lab_7_8_variant_1_artem_kornilov.cpp)**:**

**затратність ~ 3 години**

**Class Practice Task:**

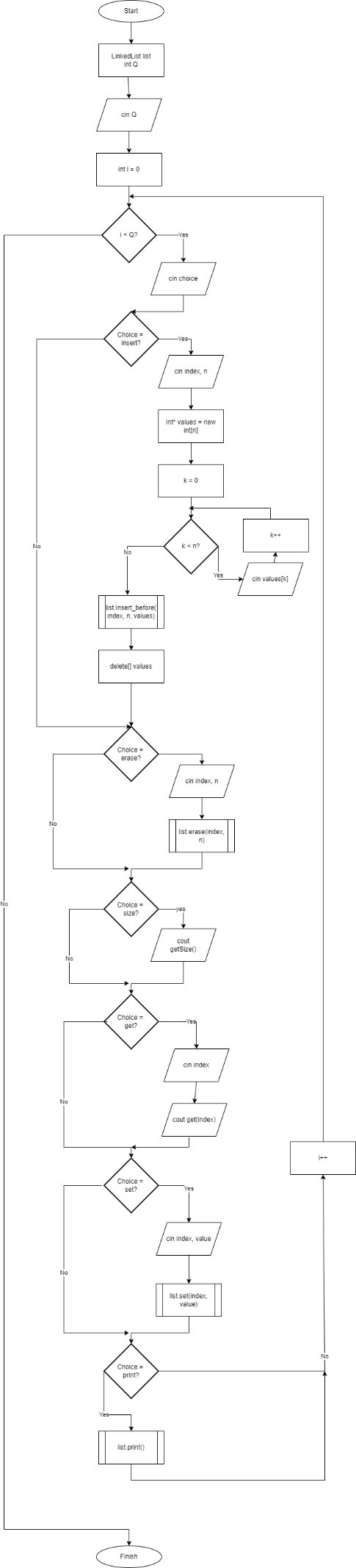
[**Practice task 1**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/practice_work_task_1_artem_kornilov.cpp)

[**Practice task 2-3**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/practice_work_task_2%2B3_john_black.cpp)

[**Practice task 4-5**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/practice_work_task_4%2B5_artem_kornilov.cpp)

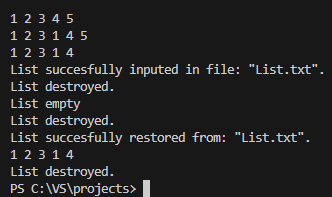
**затратність ~ 4 години**

**Block-scheme for tast** [**Algotester Lab 7-8 v1**](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/blob/epic_6_practice_and_labs_artem_kornilov/ai_11/kornilov_artem/epic6/algotester_lab_7_8_variant_1_artem_kornilov.cpp)

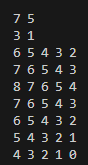


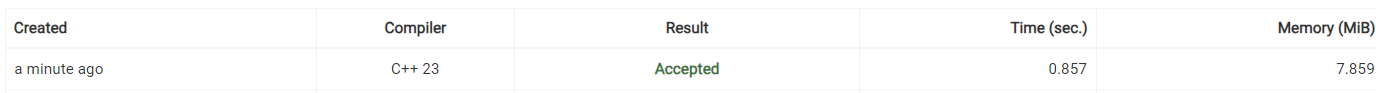
**Результати виконаних завдань, тестування та фактично затрачений час**

**VNS Lab 10v1: ~4 години**

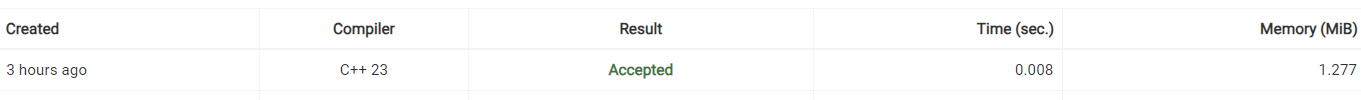


**Algotester Lab 5v3: ~2 години**



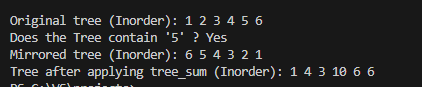
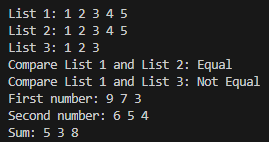


**Algotester Lab 7-8 v1: ~4 годин**



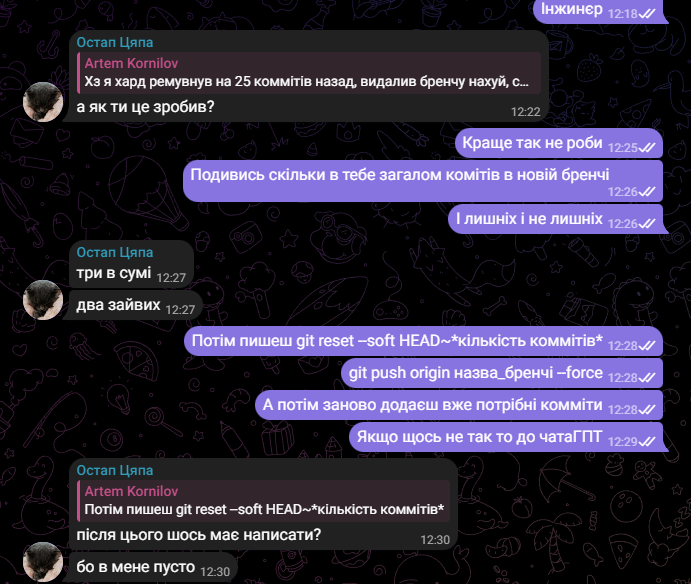
**Class Practice Task: ~4 години**





***Робота з командою:***

Налаштували Trello для Epic 6, проводили обговорення



**Висновки:**

Отже, в межах цього епіка я старався зрозуміти, що таке списки, дерева та як їх реалізовувати в коді. Практикувався з записом даних у файли, а також покращив роботу з масивами та алгоритмами.

[Посилання на pull request](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/409/files)