

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## Звіт

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

На тему: «Динамічні структури (Черга, Стек, Списки, Дерево). Алгоритми обробки динамічних структур.»

**з дисципліни:** «Основи програмування»

до:

ВНС Лабораторної Роботи № 10

Алготестер Лабораторної Роботи № 5

Алготестер Лабораторної Роботи № 7-8

Практичних Робіт до блоку № 6

**Виконав:**

Студент групи ІІІ-13

Колбасюк Данило Іванович

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

**Мета:** Ознайомитися з динамічними структурами та алгоритмами їх обробки. Реалізація у програмі мовою C++.

### **Теоретичні відомості:**

- лекції, практичні
- вказівки до лабораторних робіт ВНС
- [geeksforgeeks.org](http://geeksforgeeks.org)
- ChatGPT

### **Виконання роботи:**

- **Опрацювання завдання та вимог до роботи та середовища.**

#### **Завдання №1** Реверс списку (Reverse list)

Реалізувати метод реверсу списку.

Умови задачі:

- використовувати цілочисельні значення в списку;
- реалізувати метод реверсу;
- реалізувати допоміжний метод виведення вхідного і обернутого списків;

#### **Завдання №2** Порівняння списків

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*.

### **Завдання №3 Додавання великих чисел**

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 списку (напр.  $379 \Rightarrow 9 \rightarrow 7 \rightarrow 3$ );
- функція повертає новий список, передані в функцію списки не модифікуються.

### **Завдання №4 Віддзеркалення дерева**

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

### **Завдання №5 Записати кожному батьківському вузлу суму підвузлів**

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

### **Завдання №6 VNS Lab 10 (Variant 16)**

1. Створення списку.
2. Додавання елемента в список (у відповідності зі своїм варіантом).
3. Знищення елемента зі списку (у відповідності зі своїм варіантом).
4. Друк списку.
5. Запис списку у файл.
6. Знищення списку.
7. Відновлення списку з файлу.

Записи в лінійному списку містять ключове поле типу `*char` (рядок символів).  
Сформувати двонаправлений список. Знищити елемент із заданим ключем.  
Додати K елементів у кінець списку.

## Завдання №7 Lab 5v1

У світі Атод сестри Ліна і Рілай люблять грати у гру. У них є дошка із 8-ми рядків і 8-ми стовпців. На перетині  $i$ -го рядка і  $j$ -го стовпця лежить магічна куля, яка може світитись магічним світлом (тобто у них є 64 кулі). На початку гри деякі кулі світяться, а деякі ні... Далі вони обирають  $N$  куль і для кожної читають магічне заклиння, після чого всі кулі, які лежать на перетині стовпця і рядка обраної кулі міняють свій стан (ті що світяться - гаснуть, ті, що не світяться - загораються).

Також вони вирішили трохи Вам допомогти і придумали спосіб як записати стан дошки одним числом а із 8-ми байт, а саме (див. Примітки):

- Молодший байт задає перший рядок матриці;
- Молодший біт задає перший стовпець рядку;
- Значення біту каже світиться куля чи ні (0 - ні, 1 - так);

Тепер їх цікавить яким буде стан дошки після виконання  $N$  заклинань і вони дуже просять Вас їм допомогти.

*Вхідні дані*

У першому рядку одне число  $a$  - поточний стан дошки.

У другому рядку  $N$  - кількість заклинань.

У наступних  $N$  рядках по 2 числа  $R_i$ ,  $C_i$  - рядок і стовпець кулі над якою виконується заклинання.

*Вихідні дані*

Одне число  $b$  - стан дошки після виконання  $N$  заклинань.

## **Завдання №8 Lab78v3**

Ваше завдання - власноруч реалізувати структуру даних "Двійкове дерево пошуку".

Ви отримаєте  $Q$  запитів, кожен запит буде починатися зі слова-ідентифікатора, після якого йдуть його параметри.

Вам будуть поступати запити такого типу:

Вставка:

Ідентифікатор - insert

Ви отримуєте ціле число value - число, яке треба вставити в дерево.

Пошук:

Ідентифікатор - contains

Ви отримуєте ціле число value - число, наявність якого у дереві необхідно перевірити.

Якщо value наявне в дереві - ви виводите Yes, у іншому випадку No.

Визначення розміру:

Ідентифікатор - size

Ви не отримуєте аргументів.

Ви виводите кількість елементів у дереві.

Вивід дерева на екран

Ідентифікатор - print

Ви не отримуєте аргументів. Ви виводите усі елементи дерева через пробіл.

Вхідні дані

Ціле число  $Q$  - кількість запитів.

У наступних рядках  $Q$  запитів у зазначеному в умові форматі.

Вихідні дані

Відповіді на запити у зазначеному в умові форматі.

## **Завдання №9 Lab 5v3**

У вас є карта гори розміром  $N \times M$ . 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.

Клітинки 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.

*Вхідні дані*

У першому рядку 2 числа  $N$  та  $M$  - розміри карти

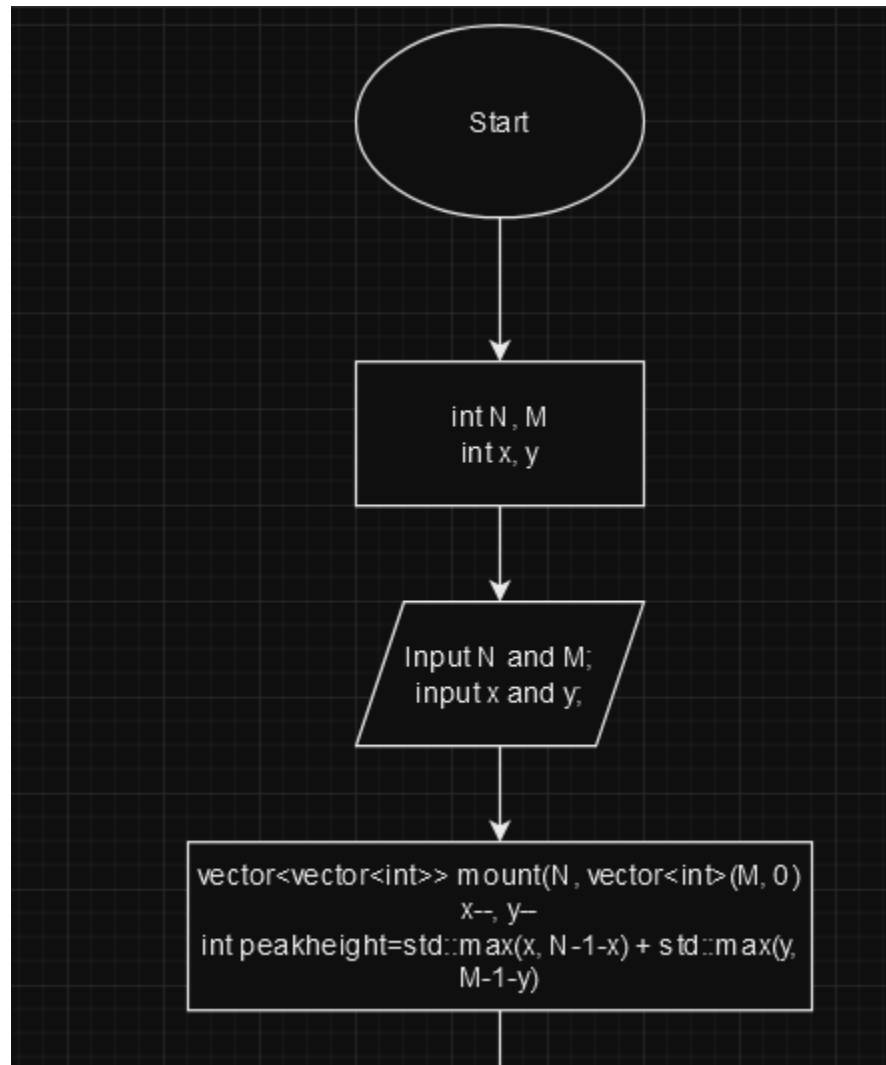
у другому рядку 2 числа  $x$  та  $y$  - координати піку гори

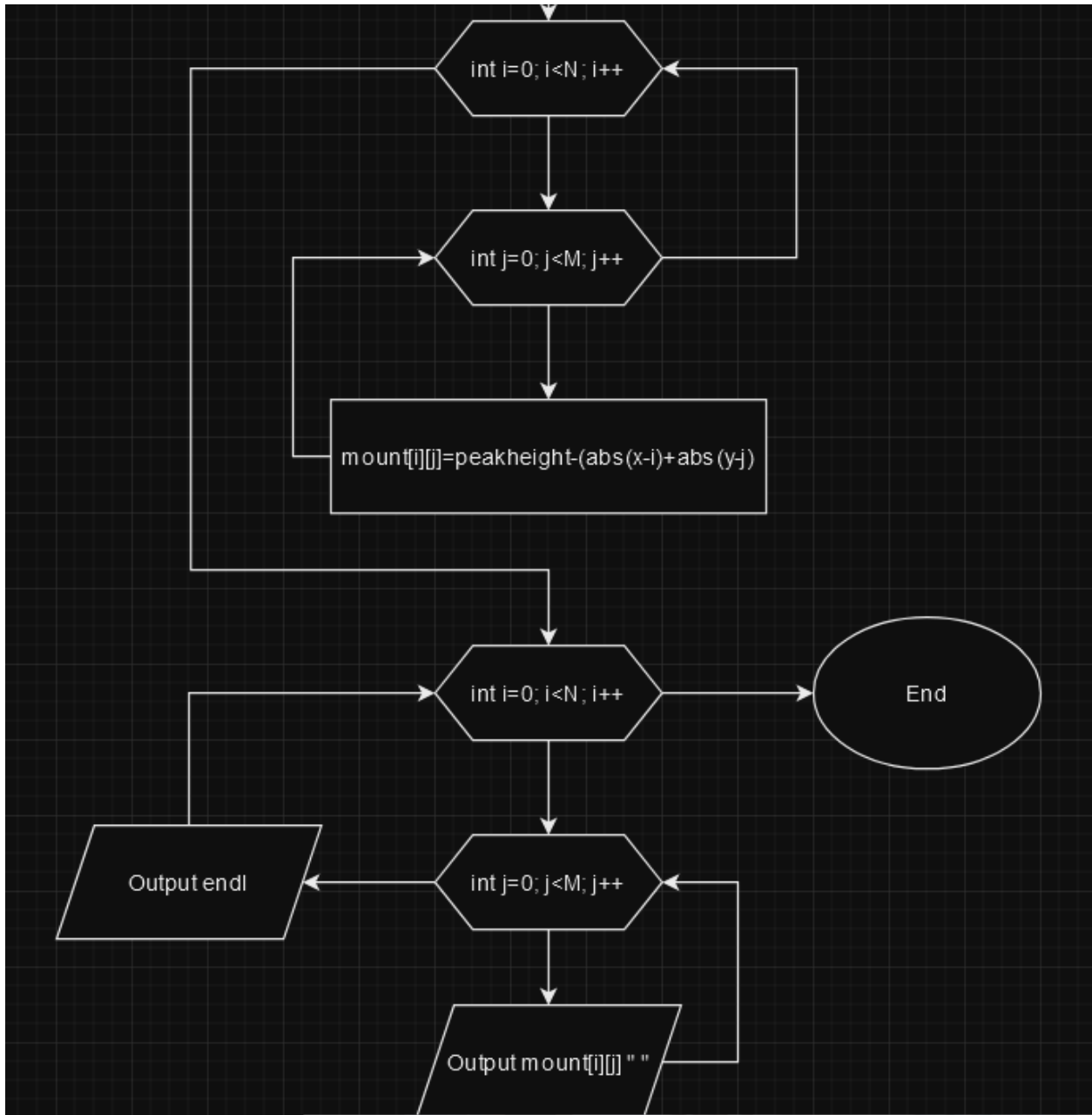
*Вихідні дані*

$N$  рядків по  $M$  елементів в рядку через пробіл - висоти карти.

- **Дизайн та планована оцінка часу виконання завдань.**

Lab 5v3





Оцінка часу:

На VNS Lab 10, practice tasks і Lab 78v3 - ~4 години на кожну.

На Lab 5v1, Lab 5v3 - 2 години на кожну.

### **- Код програми з посиланням на зовнішні ресурси**

Коди до всіх програм є у папці `eris_6` (назви файлу відповідні до номеру завдання).

## - Результати виконання завдань, тестування та фактично витрачений час

**Завдання №1** Реверс списку (Reverse list)

**Завдання №2** Порівняння списків

**Завдання №3** Додавання великих чисел

```
1 2 3 7
7 3 2 1
lists are not equal...
2 4 6 4 1
```

На ці завдання витрачено по 3 години.

**Завдання №4** Віддзеркалення дерева

**Завдання №5** Записати кожному батьківському вузлу суму підвузлів

```
2 5 6 7 8 9
9 8 7 6 5 2
2 8 6 16 8 8
```

На ці завдання витрачено по 3 години.

**Завдання №6** VNS Lab 10 (Variant 16)

```
aaaaaaaaaa dsafgijhshkj
aaaaaaaaaa dsafgijhshkj

epic_6 > ⚙ the_list.txt
1      aaaaaaaaaa dsafgijhshkj
```

На це завдання витрачено 2 години.

**Завдання №7** Lab 5v1

Lab 5v1 - Lab 5v1

C++ 23

Зараховано

```
771
4
1 1
2 1
1 2
2 2
0
```

На це завдання витрачено 2.5 години.



## Завдання №8 Lab78v3

Lab 78v3 - Lab 78v3

C++ 23

Зараховано

```
11
size
0
insert 5
insert 4
print
4 5 insert 5
print
4 5 insert 1
print
1 4 5 contains 5
Yes
contains 0
No
size
3
```

На це завдання витрачено 5 годин.

## Завдання №9 Lab 5v3

Lab 5v3 - Lab 5v3

C++ 23

Зараховано

```
3 4
2 2
1 2 1 0
2 3 2 1
1 2 1 0
```

На це завдання витрачено 2 години.

## - Командна робота

▼ **COMPLETE** 9 ... + Add Task

Name

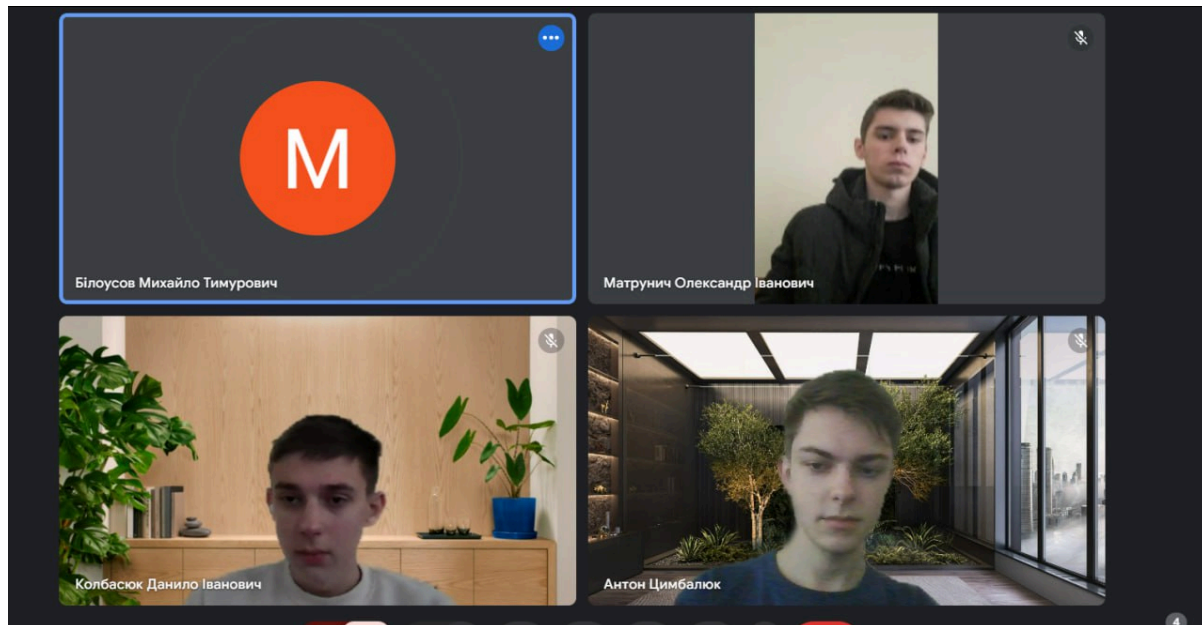
- ✓ Kobzar Artem - Epic 1 ≡ 14/14
- ✓ Danylo Kolbasiuk - Epic 1 ≡ 14/14
- ✓ Danylo Kolbasiuk - Epic 2 ≡ 6/6
- ✓ Nazar Nedosika - Epic 1 ≡ 14/14
- ✓ Alina Khodatska - Epic 1 ≡ 14/14
- ✓ Kobzar Artem - Epic 2 ≡ 5/5
- ✓ Danylo Kolbasiuk - Epic 3 ≡ 7/7
- ✓ Alina Khodatska - Epic 2 ≡ 5/5
- ✓ Kobzar Artem - Epic 3 ≡ 7/7

+ Add Task

▼ **IN PROGRESS** 7 ... + Add Task

Name

- Danylo Kolbasiuk - Epic 4 ≡ 5/7
- Tofan Max - Epic 1 ≡ 0/14
- Nazar Nedosika - Epic 2 ≡ 4/5
- Tofan Max - Epic 2 ≡ 0/5
- Tofan Max - Epic 3 ≡ 0/7
- Nazar Nedosika - Epic 3 ≡ 4/7
- Alina Khodatska - Epic 3 ≡ 0/7



**Висновок:** в цьому епіку я ознайомився з динамічними структурами, а також навчився їх реалізовувати за допомогою класів у C++.

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/598](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/598)