

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:
Практичних Робіт до блоку № 6

Виконав:
Студент групи ШІ-13
Цимбалюк Антон Едуардович

Львів 2024

Тема роботи:

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

Мета роботи:

Навчитися працювати з динамічними структурами та писати алгоритми для їх обробки.

Теоретичні відомості:

1. Теми, необхідні для виконання роботи:

- Однозв'язний та двозв'язний списки.
- Дерево.

2. Джерела використані для ознайомлення з вищезазначеними темами:

- Дивився деякі моменти в чата GPT та інтернеті.
- З усією іншою інформацією я був знайомий з власного досвіду.

Виконання роботи:

Опрацювання завдання та вимог до програми та середовища

Завдання №1 - Class Practice Work Task 1-3 – Зв'язаний список

Задача

1. 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;

2. Реалізувати метод порівняння списків `bool compare(Node *h1, Node *h2);`

Умови задачі:

- використовувати цілочисельні значення в списку;
- реалізувати функцію, яка ітеративно проходить 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**.

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

Завдання №2 - Class Practice Work Task 4-5 – Бінарні дерева

Задача

4. Реалізувати метод відзеркалення 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

5. Реалізувати метод запису 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

Завдання №3 – VNS lab 10 варіант 4

Задача

Написати програму, у якій створюються динамічні структури й виконати

їхню обробку у відповідності зі своїм варіантом.

Для кожного варіанту розробити такі функції:

- 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 int. Сформувати однонаправлений список. Знищити з нього елемент із заданим номером, додати K елементів, починаючи із заданого номера;

Завдання №4 – Algotester lab 5 variant 2

Задача

пустелі існує незвичайна печера, яка є двохвимірною. Її висота це N, ширина - M.

Всередині печери є пустота, пісок та каміння. Пустота позначається буквою O , пісок S і каміння X;

Одного дня стався землетрус і весь пісок посипався вниз. Він падає на найнижчу клітинку з пустотою, але він не може пролетіти через каміння.

Ваше завдання сказати як буде виглядати печера після землетрусу.

Input

У першому рядку 2 цілих числа N та M - висота та ширина печери
У N наступних рядках стрічка row_i яка складається з N цифер - i -й рядок матриці, яка відображає стан печери до землетрусу.

Завдання №5 – Algotester lab 7-8 variant 3

Задача

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.
Реалізувати використовуючи перегрузку оператора <<

Завдання №6 – Self practice Algotester Graph on A plane

You are given n unique points (x_i, y_i) on a two-dimensional plane.

A weighted undirected graph G with real-valued weights is good if it satisfies the following conditions.

- G contains n vertices numbered 1 through n .
- For each pair of vertices (i, j) , the shortest-path weight between vertices i and j in G equals the Euclidean distance between points (x_i, y_i) and (x_j, y_j) .

Find the minimum number of edges in a good graph.

Input

The first line contains an integer n – the number of points.

The following n lines contain two integers x_i and y_i – coordinates of the points.

.

Дизайн та планована оцінка часу виконання завдань

Завдання №1 - Class Practice Work Task 1-3 – Зв’язаний список

розраховував, що це завдання займе 1 годину

Завдання №1 - Class Practice Work Task 4-5 – Бінарне дерево

розраховував, що це завдання займе 1 годину

Завдання №2 – VNS lab 10 варіант 4

розраховував, що це завдання займе 1 годину

Завдання №3 – Algotester lab 5 variant 2

розраховував, що це завдання займе 20 хвилин

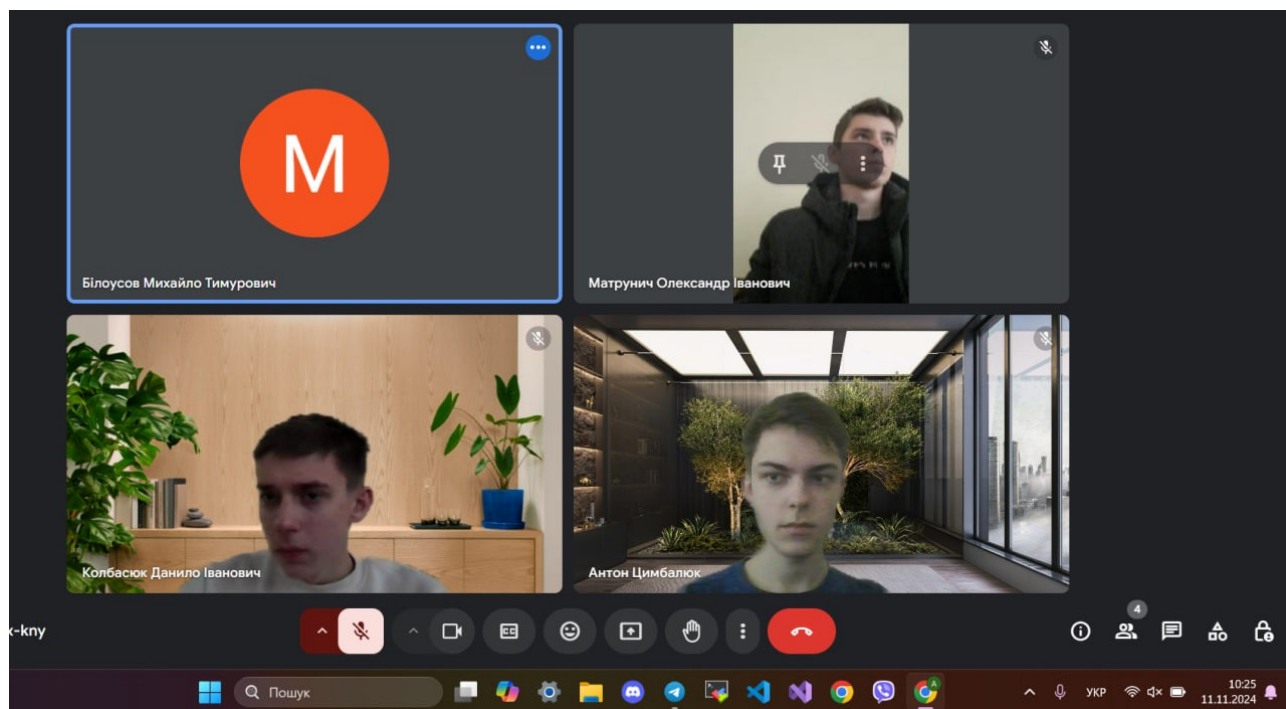
Завдання №4 – Algotester lab 7-8 variant 3

розраховував, що це завдання займе 1 годину

Завдання №5 – Self practice Algotester graph on a plane

розраховував, що це завдання займе 2 години

Скрін зустрічі нашої команди



Результати виконаних завдань, тестування та фактично затрачений час

Завдання №1 - Class Practice Work Task 1-3 – Зв'язаний список

```
Enter number of elements you want to add: 5
Enter values: 1 2 3 4 5
Reversed list:
5 4 3 2 1
List is reversed back
Enter number of elements you want to add into second list: 5
Enter values: 0 9 3 2 2
List are equal : false
List1 + List2 =
0 1 2 2 3 3 4 5 9
```

Зайняло часу – 1 година

Завдання №2 - Class Practice Work Task 4-5 – Бінарне дерево

```
3 6 10 11 15
Mirrored tree:
15 11 10 6 3
Sums of subnodes of a tree:
Sum of subnodes for node of value 6 = 0
Sum of subnodes for node of value 3 = 6
Sum of subnodes for node of value 15 = 0
Sum of subnodes for node of value 11 = 15
Sum of subnodes for node of value 10 = 35
```

Зайняло часу – 1 година

Завдання №3 – VNS lab 10 варіант 4


```

[1] - to add elements
[2] - to remove elements
[3] - save list to file
[4] - get list from file
[5] - to show list
[0] - to exit
1
Index at which to add elements: 1
Enter number of elements you want to add: 5
Enter values: 1 2 3 4 5
[1] - to add elements
[2] - to remove elements
[3] - save list to file
[4] - get list from file
[5] - to show list
[0] - to exit
5
1 2 3 4 5

```

Зайняло часу – 30 хвилин

Завдання №4 – Algotester lab 5 variant 2

```

4 4
S00S
X000
S000
0000
-----
S000
X000
0000
S00S

```

```

4 3
S00
00X
000
00S
-----
000
00X
000
S0S

```

Зайняло часу – 30 хвилин

Завдання №5 – Algotester lab 7-8 variant 3

```

3
insert 1
insert 2
print
1 2

```

```

6
print

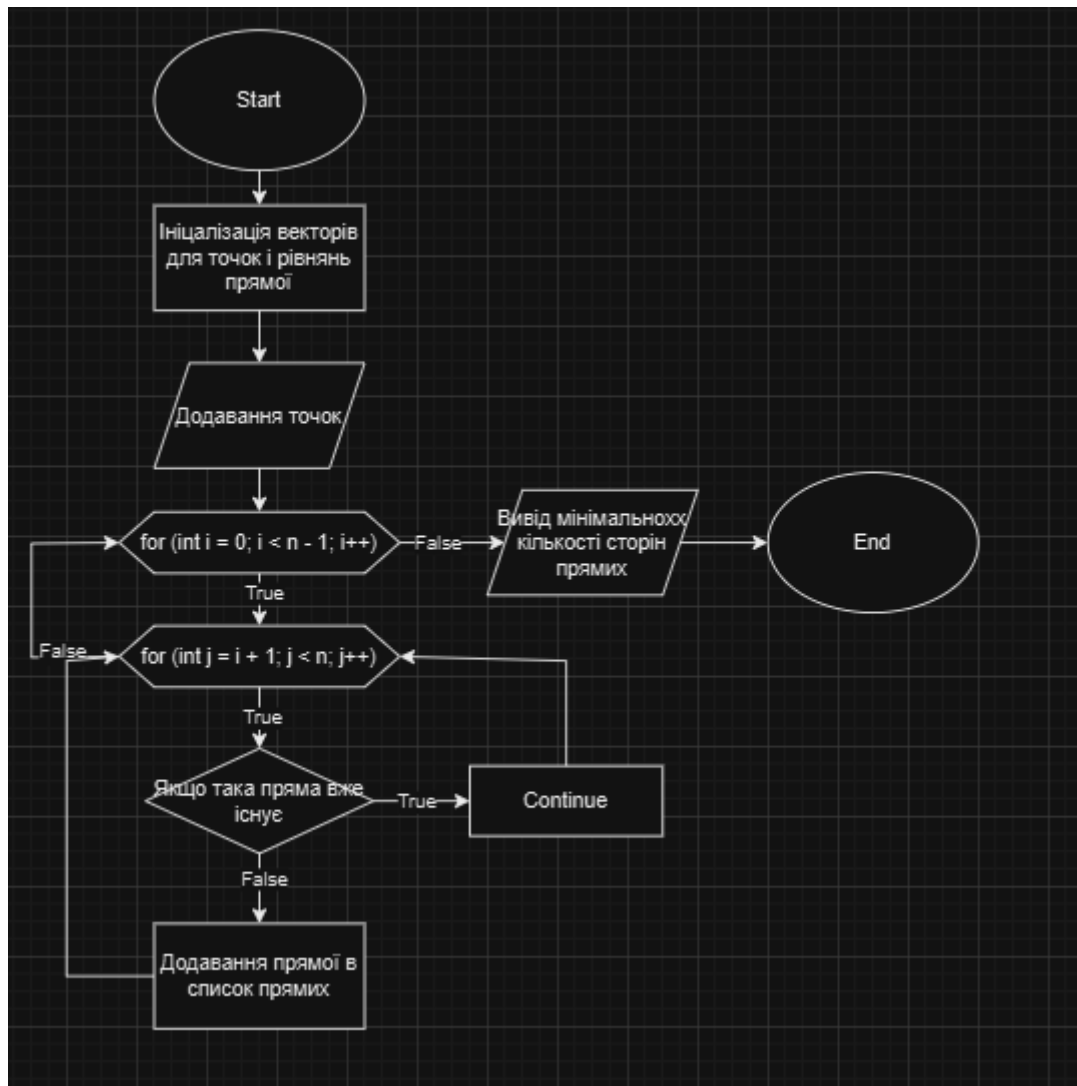
insert 2
insert 2
insert 3
remove 2
print
3

```

Зайняло часу – 30 хвилин

Завдання №6 – Self practice graph on a plane

```
4
0 0
4 3
4 2
0 4
6
```



Зайняло часу – 1.5 години

Висновок: Я навчився працювати з динамічними структурами та писати алгоритми для їх обробки.