

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 ШІ-13
Бойко Роман Андрійович

Львів 2024

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

Мета роботи: Навчитися користуватися різними видами динамічних структур.

Теоретичні відомості:

- Бінарне дерево пошуку
- Списки
- Черга

Джерела:

- [С++ • Теорія • Урок 144 • ADT • Бінарне дерево](#)
- [С++ • Теорія • Урок 139 • ADT • Однозв'язний список](#)
- [С++ • Теорія • Урок 140 • ADT • Двозв'язний список](#)
- [С++ • Теорія • Урок 141 • ADT • Стек](#)
- [С++ • Теорія • Урок 142 • ADT • Черга](#)

Виконання роботи

Особистий варіант - VNS Lab 10 - 1, Algotester Lab 7-8 - 2,3, Algotester Lab 5 - 2

Завдання 1: Algotester Lab 5

В пустелі існує незвичайна печера, яка є двохвимірною. Її висота це N, ширина - M. Всередині печери є пустота, пісок та каміння. Пустота позначається буквою , 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

Вивід в терміналі:

```
5 5
SSOSS
00000
S00XX
0000S
00S00

00000
000SS
000XX
S0000
SSSOS
```

Час виконання завдання ~ 30 хв

Завдання 2: Self Practice Task Algotester Lab 7-8 variant 3

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 <<

Вивід в терміналі:

```
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
```

Час виконання завдання ~ 2 години

Завдання 3: VNS Lab 10

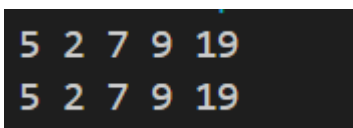
Написати програму, у якій створюються динамічні структури й виконати їхню обробку у відповідності зі своїм варіантом.

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 елемент із заданим номером;

Вивід в терміналі:



```
5 2 7 9 19
5 2 7 9 19
```

Час виконання завдання ~ 1 година

Завдання 4: Practice Task 1-3

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

Результат виконання програми:

```
The list is empty!

List before reverse: 5 10 7 0 6 4
List after reverse: 4 6 0 7 10 5
Lists are not equal!
Sum: 0 1 2 1 0 1 1
```

Час виконання завдання ~ 2.5 години

Завдання 5: Practice Task 4-5

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

Результат виконання програми:

```
Normal tree: 0 1 3 4 5 6 7 9
Sum at node 0: 0
Sum at node 5: 5
Sum at node 6: 11
Sum at node 4: 15
Sum at node 3: 18
Sum at node 9: 9
Sum at node 7: 34
Sum at node 1: 35
```

Час виконання завдання ~ 1.5 години

Завдання 6: Algotester Lab 7-8 Variant 2

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

- Вставка: Ідентифікатор - insert Ви отримуєте ціле число index елемента, на місце якого робити вставку. Після цього в наступному рядку рядку написано число N - розмір масиву, який треба вставити. У третьому рядку N цілих чисел - масив, який треба вставити на позицію index.
- Видалення: Ідентифікатор - erase Ви отримуєте 2 цілих числа - index, індекс елемента, з якого почати видалення та n - кількість елементів, яку треба видалити.
- Визначення розміру: Ідентифікатор - size Ви не отримуєте аргументів. Ви виводите кількість елементів у динамічному масиві.
- Визначення кількості зарезервованої пам'яті: Ідентифікатор - capacity Ви не отримуєте аргументів. Ви виводите кількість зарезервованої пам'яті у динамічному масиві. Ваша реалізація динамічного масиву має мати фактор росту (Growth factor) рівний 2.
- Отримання значення i-го елемента Ідентифікатор - get Ви отримуєте ціле число - index, індекс елемента. Ви виводите значення елемента за індексом. Реалізувати використовуючи перегрузку оператора []
- Модифікація значення i-го елемента Ідентифікатор - set Ви отримуєте 2 цілих числа - індекс елемента, який треба змінити, та його нове значення. Реалізувати використовуючи перегрузку оператора []
- Вивід динамічного масиву на екран Ідентифікатор - print Ви не отримуєте аргументів. Ви виводите усі елементи динамічного масиву через пробіл. Реалізувати використовуючи перегрузку оператора <<

Результат виконання програми:

```
12
size
0
capacity
1

insert 0 2
100 100

size
2
capacity
4

insert 0 2
102 102

size
4
capacity
8

insert 0 2
103 103

size
6
capacity
8

print
103 103 102 102 100 100
```

Час виконання завдання ~ 3 години

Завдання 7: Self Practice Task Algotester Lab 3 variant 1

Ви з'явилися у світі під назвою Атод посеред Пустелі Безправ'я. Так сталося, що Ви попали саме в той час і місце, де ведеться битва між чаклункою Ліною і темними силами, які хочуть знищити цей світ. На жаль, трапилась халепа, бо деякі слова із книги чар були пошкоджені під час битви. Одне таке слово можна відновити виконавши ритуал зцілення над пошкодженими буквами. Ритуал зцілення можна виконати на всіх підряд розташованих пошкоджених буквах. Вам не залишається нічого іншого як допомогти Ліні відновити ці слова і сказати скільки мінімально треба провести таких ритуалів, щоб прочитати одне з наймогутніших у цьому світі заклять - Поневолення Дракона!

Результат виконання програми:

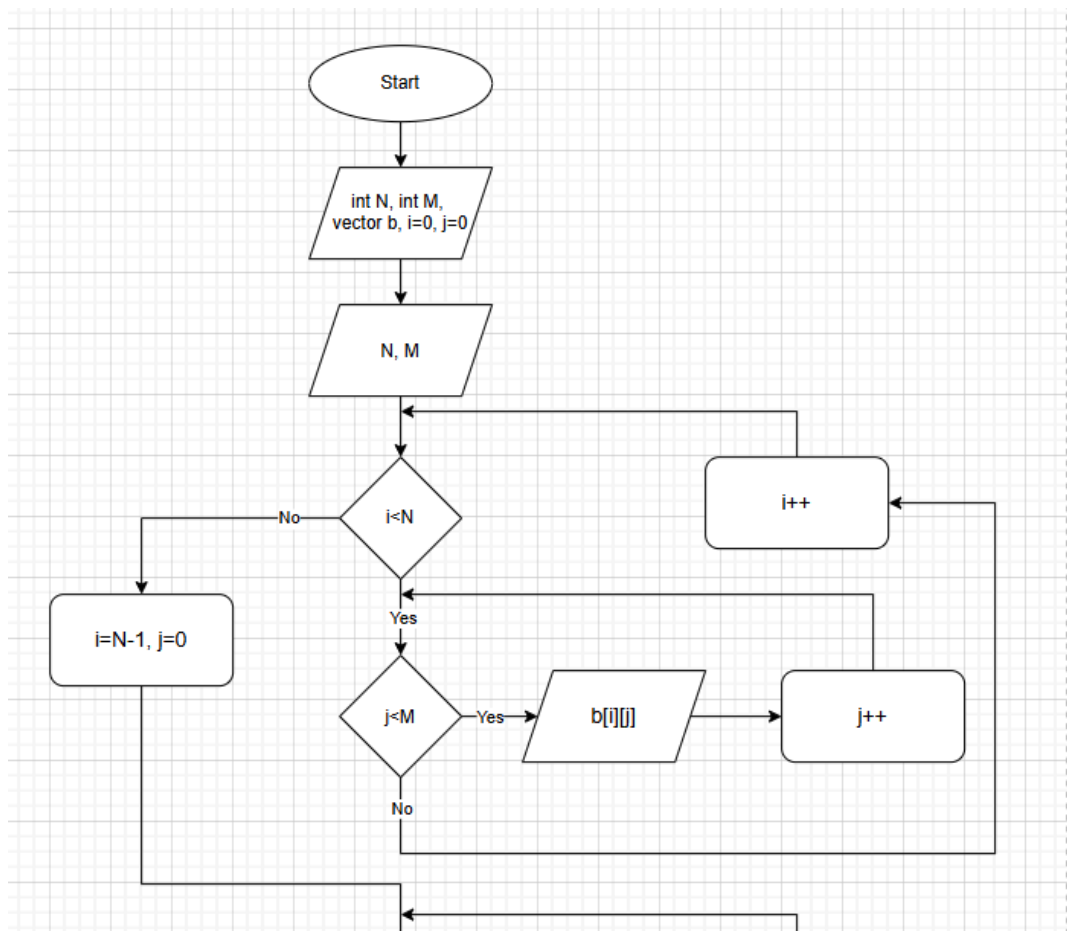
```

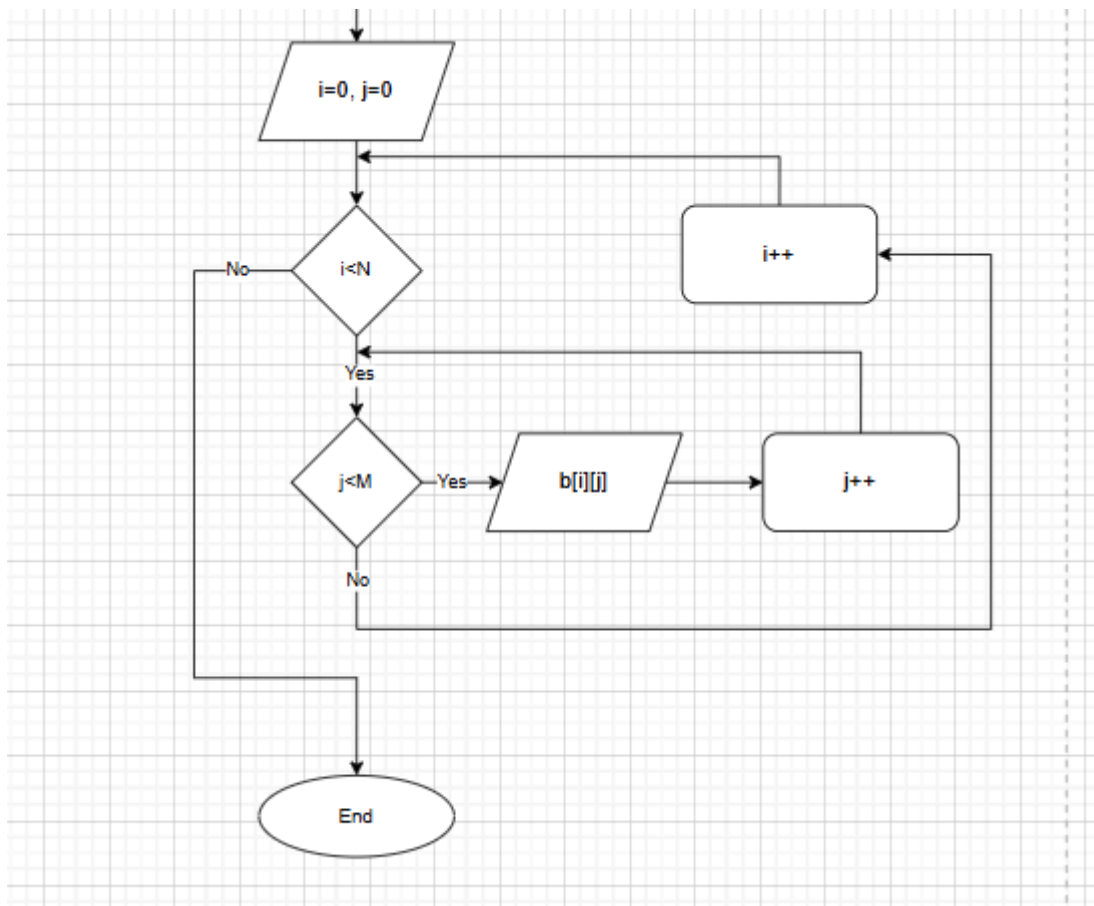
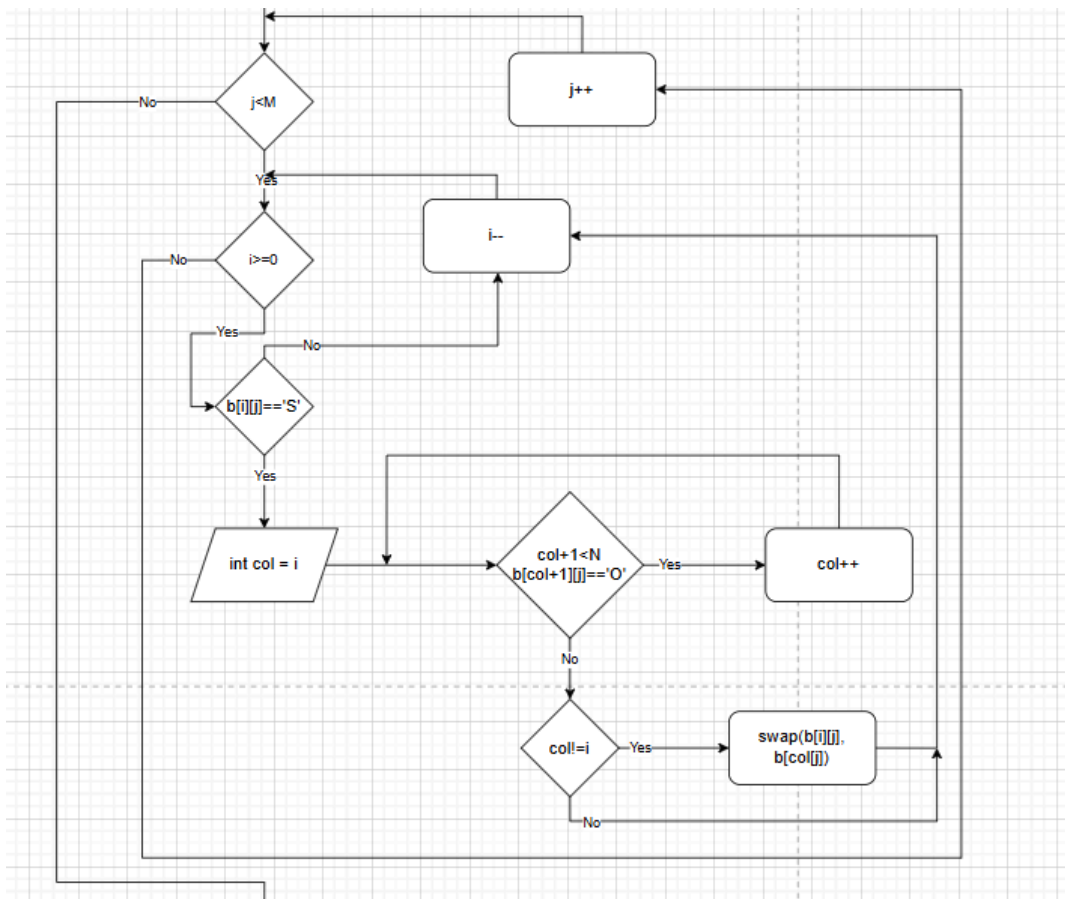
8
d#rkness b#yond twilight
crimson beyond blood that flows
buried in the stream of time is where your power grows
i pledge myself to conquer all the foes who stand
before the mighty gift bestowed in my unworthy hand
let the fools who stand before me be destroyed
by the power you and i possess
dr#gon sl#ve
4

```

Час виконання завдання ~ 15 хвилин

Блок-схема до задачі Algotester Lab 5





Висновок:

У цьому епіку я навчився працювати з різними видами динамічних структур