

Міністерство освіти і науки України
Національний технічний університет України «Київський
політехнічний
інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт
з лабораторної роботи № 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»
«ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ »
Варіант 32

Виконав студент ІП-13, Черкасов Станіслав Олексійович

(шифр, прізвище, ім'я, по батькові)

Перевірів Вечеровська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Лабораторна робота 9

ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Варіант 32

32	Задано матрицю дійсних чисел $A[m,n]$. При обході матриці по стовбцям знайти в ній перший додатний елемент X і його місцезнаходження. Порівняти X з середньоарифметичним значенням елементів над побічною діагоналлю.
----	--

Постановка задачі

- 1) Задамо матрицю MATRIX довільного розміру $X*Y$ з дійсних значень. Ініціюємо заповнення випадковими числами.
- 2) Напишемо функцію для знаходження першого додатного елементу проходженням по стовпчиках матриці. (FP)
- 3) Напишемо функцію для знаходження середнього арифметичного значення елементів над побічною діагоналлю (AASD) (сума індексів елементів побічної діагоналі завжди дорівнює $X-1$ (кількість рядків - 1). Отже, сума індексів елементів над нею завжди буде менше $X-1$.
- 4) Порівняємо два отриманих значення та виведемо результат порівняння: “більше”, якщо $FP > AASD$, “рівно”, якщо $FP = AASD$, “менше”, якщо $FP < AASD$.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
двовимірний масив	масив $X*Y$ дійсних чисел	MATRIX	Початкове дане
кількість рядків	Натуральне	ROWS	Початкове дане
кількість стовпчиків	Натуральне	COLUMNS	Початкове дане
лічильник циклу	Натуральне	I	Проміжне дане
лічильник циклу	Натуральне	J	Проміжне дане
перший додатний елемент масиву	Натуральне	FP	Проміжне дане
сума елементів над поб. діаг.	Ціле	SUM	Проміжне дане
кількість елементів над поб. діаг.	Ціле	N	Проміжне дане
Середнє арифметичне елементів над поб. діаг.	Ціле	AASD	Проміжне дане
Результат порівняння	Строкове	ANS	Кінцеве дане

Rand(p, q) - Функція, що повертає випадкове ціле значення між P та Q не включаючи P та Q

Розв'язання

Програмні специфікації запишемо у формі псевдокоду та у вигляді блок-схеми.

А) Функція MATRIX_INIT():

MATRIX_INIT(M, X, Y)

для I від 0 до X:

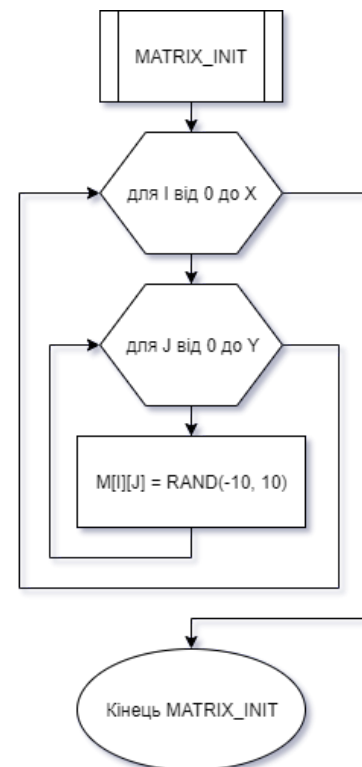
для J від 0 до Y:

$M[I][J] = \text{rand}(-10, 10)$

все для

все для

Кінець MATRIX_INIT



Б) Функція FIRST_POSITIVE():

FIRST_POSITIVE(M, X, Y)

для J від 0 до Y:

для I від 0 до X:

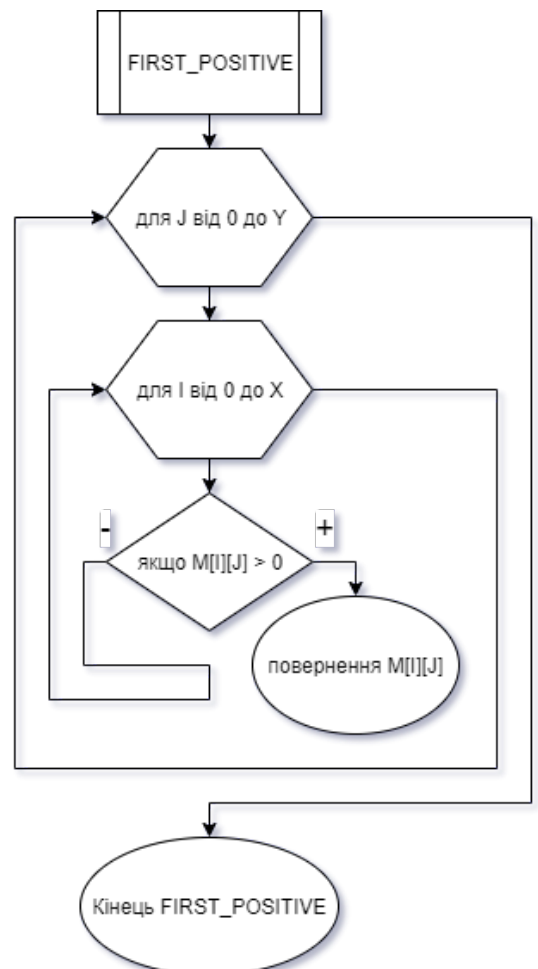
якщо $M[I][J] > 0$:

повернути $M[I][J]$

все для

все для

Кінець FIRST_POSITIVE



В) Функція

AVERAGE_ABOVE_SEC_DIAG():

AVERAGE_ABOVE_SEC_DIAG(M, X, Y)

SUM = 0

N = 0

для I від 0 до X:

для J від 0 до Y:

якщо I + J < X - 1:

SUM += M[I][J]

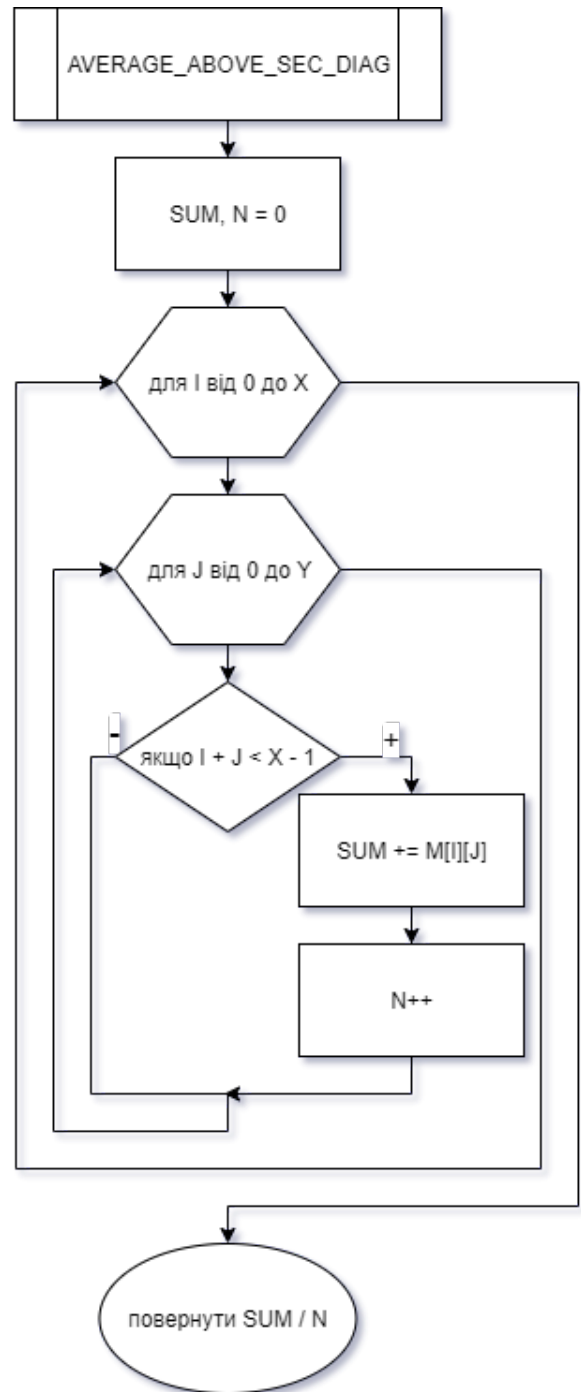
N ++

все для

все для

повернути SUM / N

Кінець AVERAGE_ABOVE_SEC_DIAG



Б) MAIN

Псевдокод

початок

введення ROWS, COLUMNS

MATRIX = INT

[ROWS][COLUMNS]

MATRIX_INIT(MATRIX, ROWS,
COLUMNS)

FP = FIRST_POSITIVE(MATRIX,
ROWS, COLUMNS)

AASD =

AVERAGE_ABOVE_SEC_DIAG(MATRIX,
ROWS, COLUMNS)

якщо FP > AASD:

ANS = “більше”

інакше якщо FP < AASD:

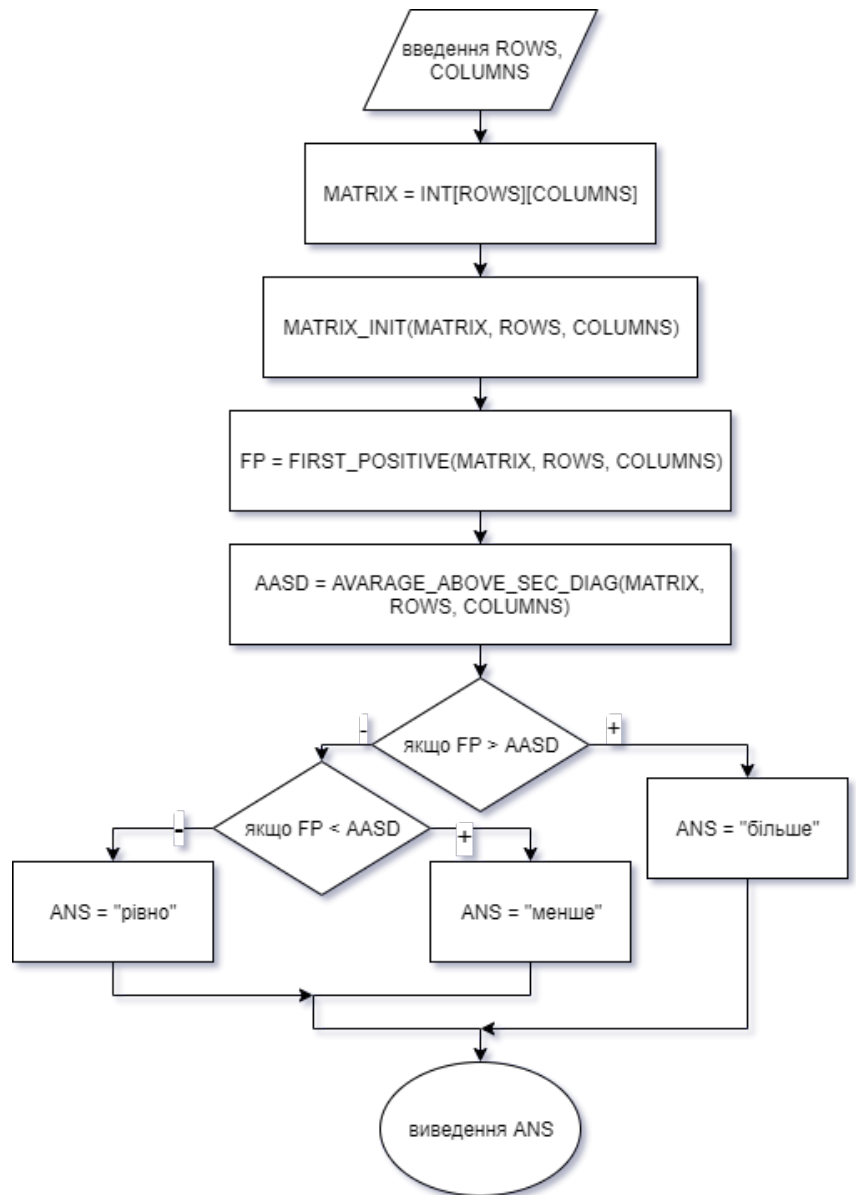
ANS = “менше”

інакше:

ANS = “рівно”

виведення ANS

кінець



Випробування Алгоритму:

```
M is an XxY matrix. Define X: 5
Define Y: 7

The matrix:
 5 6-7 6-4 0-3
-4 7-4-4-2-4 1
-2 3-8-1 0 8-3
 8 2 6 8 0-4-4
 4-1-9 0 5 7-3

First positive: 5
Average above secondary diagonal: 1.8
5 > 1.8
```

```
M is an XxY matrix. Define X: 8
Define Y: 4

The matrix:
-1-2 2-7
 4 3 7-5
 1 0 6-3
-9-3 1-6
 8 8 2 8
 5 7 8-4
 8 1 1 8
-2 5 6-6

First positive: 4
Average above secondary diagonal: 1.18182
4 > 1.18182
```

```
M is an XxY matrix. Define X: 2
Define Y: 7

The matrix:
 2-5 7-7 1-8-7
-1-1-8-3-5 1 5

First positive: 2
Average above secondary diagonal: 2
2 = 2
```

Висновок

Під час виконання цієї лабораторної роботи я вдосконалив навички написання математичної моделі, праці з блок схемами та випробування алгоритму.

Дослідив алгоритми обходу масивів, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій.