

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 № 8
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»

«ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ »
Варіант 12

Виконала студентка ІП-15 Коваленко Марія Олександрівна
(шифр, прізвище, ім'я, по батькові)
Перевірила Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Лабораторна робота 7

Дослідження алгоритмів обходу масивів

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

Індивідуальне завдання

Варіант 12

Завдання

12. Задано матрицю дійсних чисел $A[m,n]$. При обході матриці по стовпчиках знайти в ній останній мінімальний елемент і його місцезнаходження. Обміняти знайдене значення X з елементом першого рядка.

Постановка задачі

Задамо матрицю.

Знайдемо x обходом по стовпцям

Обміняємо знайдене значення x з елементом першого рядка.

Побудова математичної моделі

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
лічильник	цілий	i	проміжні дані
лічильник	цілий	k	проміжні дані
розмірність матриці і масиву	цілий	m	проміжні дані
розмірність матриці	цілий	n	проміжні дані
вихідна матриця	дійсний	matrix	проміжні дані
останнє найменше значення	дійсний	x	проміжні дані, вихідні дані
індекс рядка зі змінною x	цілий	row	проміжні дані, вихідні дані
індекс стовпчика зі змінною x	цілий	col	проміжні дані, вихідні дані
допоміжна змінна для обходу по стовпцям	цілий	index	проміжні дані

будемо використовувати функцію $\text{abs}(a)$, яка повертає модуль числа

Для обходу по стовпцям я використала формулу
$$\text{index} = (m - k - 1) * (i \% 2) + k * \text{abs}(i \% 2 - 1)$$

вона складається з двох доданків, один із яких завжди нуль, а інший — індекс рядка елемента

наприклад для стовпця з індексом 0 ненульовим буде доданок $k * \text{abs}(i \% 2 - 1)$ тому стовпець буде переглядатися згори вниз

а от для стовпця з індексом 1 ненульовим буде доданок $(m - k - 1) * (i \% 2)$ тому стовпець буде переглядатися знизу вгору

Розв'язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо ініціалізування матриці

Крок 3. Деталізуємо знаходження необхідного масиву

Крок 4. Деталізуємо сортування необхідного масиву

Псевдокод

початок

введення m, n

повторити для i від 0 до m

повторити для k від 1 до n

$\text{matrix}[i, k] := \text{rand}(1, 1000) + \text{rand}(1, 100) * 0.01$

кінець циклу

кінець циклу

$x = \text{matrix}[0][0]$

$\text{row} = 0$

$\text{col} = 0$

$\text{index} = 0$

повторити для i від 0 до n

повторити для k від 0 до m

```
index = (m - k - 1) * (i % 2) + k * abs(i % 2 - 1);  
якщо x >= matrix[index][i]  
то      x = matrix[index][i];  
        row = index;  
        col = i;
```

кінець циклу

кінець циклу

```
matrix[row][col] = matrix[0][col];
```

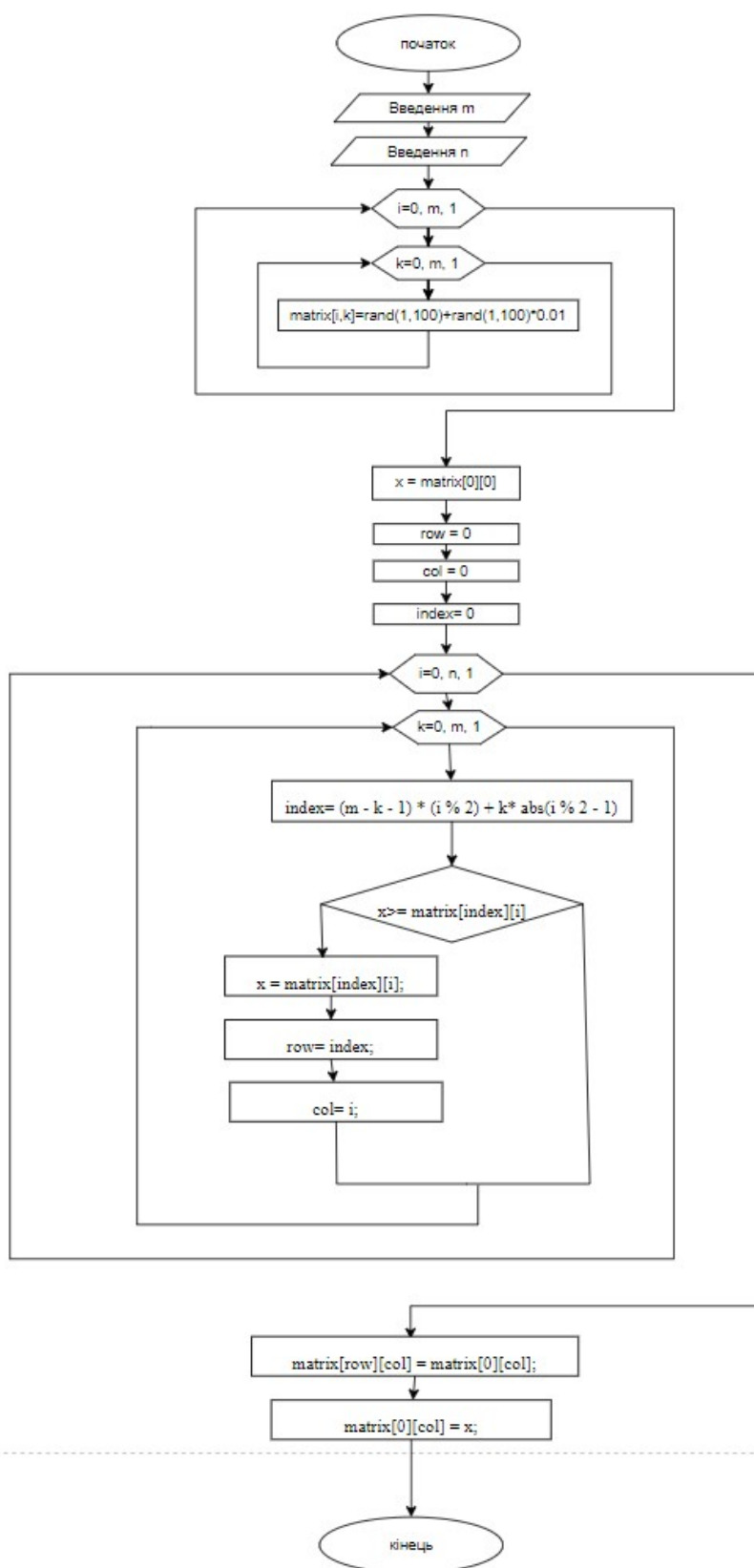
```
matrix[0][col] = x;
```

кінець

*в псевдокод та блоксхему не включені дії для виводу матриць та повернення пам'яті

Блок-схема

Основи програмування – 1. Алгоритми та структури даних



Код

```

7
8 int main()
9 {
10     int m;
11     int n;
12     cin >> m;
13     cin >> n;
14
15     float **matrix = new float*[m];
16
17     for (int i = 0; i < m; i++)
18     {
19         matrix[i] = new float[n];
20         for (int k = 0; k < n; k++)
21         {
22             matrix[i][k] = rand()%100 + (rand() % 100) * 0.01;
23         }
24     }
25
26     //вивід отриманої матриці
27     for (int i = 0; i < m; i++) {
28         for (int k = 0; k < n; k++)
29         {
30             cout << setw(5) << matrix[i][k] << " ";
31         }
32         cout << "" << endl;
33     }
34     cout << "" << endl;
35
36     float x = matrix[0][0];
37     int row = 0;
38     int col = 0;
39     int index = 0;
40
41     for (int i = 0; i < n; i++)
42     {
43         for (int k = 0; k < m; k++)
44         {
45             index = (m - k - 1) * (i % 2) + k * abs(i % 2 - 1);
46             if (x >= matrix[index][i]) {
47                 x = matrix[index][i];
48                 row = index;
49                 col = i;
50             }
51         }
52     }
53     matrix[row][col] = matrix[0][col];
54     matrix[0][col] = x;
55
56     //вивід отриманої матриці
57     for (int i = 0; i < m; i++) {
58         for (int k = 0; k < n; k++)
59         {
60             cout << setw(5) << matrix[i][k] << " ";
61         }
62         cout << "" << endl;
63     }
64
65     //повернення пам'яті
66     for (int i = 0; i < m; i++)
67     {
68         delete[] matrix[i];
69     }
70     delete[] matrix;
71
72
73
74
75
76

```

Тестування

```
Вибрати консоль отладки microsoft visual studio
5
7
41.67    34 69.24 78.58 62.64  5.45 81.27
61.91 95.42 27.36 91.04  2.53 92.82 21.16
18.95 47.26 71.38 69.12 67.99 35.94  3.11
22.33 73.64 41.11 53.68 47.44 62.57 37.59
23.41 29.78 16.35 90.42 88.06 40.42 64.48

41.67    34 69.24 78.58  2.53  5.45 81.27
61.91 95.42 27.36 91.04 62.64 92.82 21.16
18.95 47.26 71.38 69.12 67.99 35.94  3.11
22.33 73.64 41.11 53.68 47.44 62.57 37.59
23.41 29.78 16.35 90.42 88.06 40.42 64.48

C:\Users\Ria\Desktop\flyyyyyyyyyyy\kpi_study\1 курс\Алгоритми та ф
```

Висновок:

Ми дослідили алгоритми обходу масивів, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій.