Циклічні алгоритми

к.т.н., доцент кафедри прикладної математики
Рижа Ірина Андріївна

Про що ця лекція???

- Викладемо особливості організації та роботи із багатовимірними масивами.
- Розглянемо приклади алгоритмів деяких задач.

Багатовимірні масиви

Двовимірний масив

– це впорядкована сукупність однотипних елементів, у якій для знаходження елементів масиву необхідно **два** індекси.

- Такі масиви зручно представляти у вигляді таблиці.

Тривимірний масив

– це впорядкована сукупність однотипних елементів, у якій для знаходження елементів масиву необхідно **три** індекси.
Наприклад,

- якщо сторінку тексту трактувати, як двовимірний масив символів, то всю книгу можна трактувати, як тривимірний масив символів;
- перший індекс відповідає за номер сторінки, другий – за номер рядка на сторінці, третій – за номер символу у рядку.

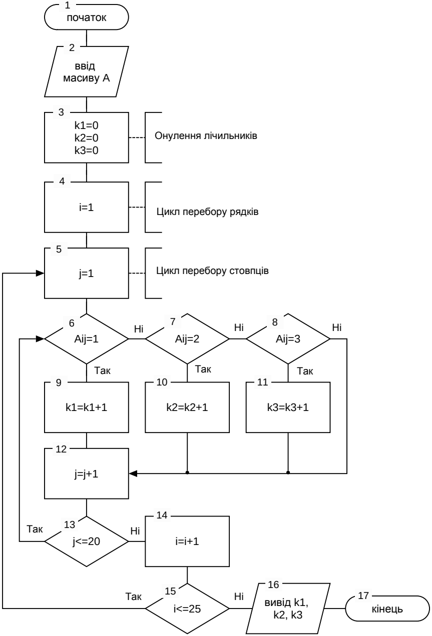
Приклад 1.

Розглянемо зал засідань у якому 25 рядів по 20 місць. Місця зайняті делегатами, які приймають рішення по запропонованих їм пропозиціям шляхом голосування і відповідна резолюція приймається після підрахунку кількості голосів. Кожен з делегатів висловлює свою думку, голосуючи **“за”**, **“утримуюсь”**, або **“проти”**. Побудувати алгоритм підведення підсумків голосування, враховуючи, що окремі місця можуть бути не зайняті делегатами.

- A – “таблиця голосування” – двовимірний масив, який складається з 25 рядків і 20 стовпців;
- i, $1 \leqslant i \leqslant 25$ – перша індексна змінна, яка задає номер рядка;
- j, $1 \leqslant j \leqslant 20$ – друга індексна змінна, яка задає номер стовпця.

$i \backslash j$	1	2	$\cdots$	20
1				
2				
$\vdots$				
25				

$A_{i,j} = 0$ – делегат відсутній;
$A_{i,j} = 1$ – “за”;
$A_{i,j} = 2$ – “утримався”;
$A_{i,j} = 3$ – “проти”;
$k_1,\ k_2,\ k_3$ – лічильники.


1
початок
2
ввід
масиву A
3
k1=0
k2=0
k3=0
Онулення лічильників
4
i=1
Цикл перебору рядків
5
j=1
Цикл перебору стовпців
6
Aij=1
Ні
7
Aij=2
Ні
8
Aij=3
Ні
Так
Так
Так
9
k1=k1+1
10
k2=k2+1
11
k3=k3+1
12
j=j+1
13
j<=20
Так
Ні
14
i=i+1
15
i<=25
Так
Ні
16
вивід k1,
k2, k3
17
кінець

Приклад 2.

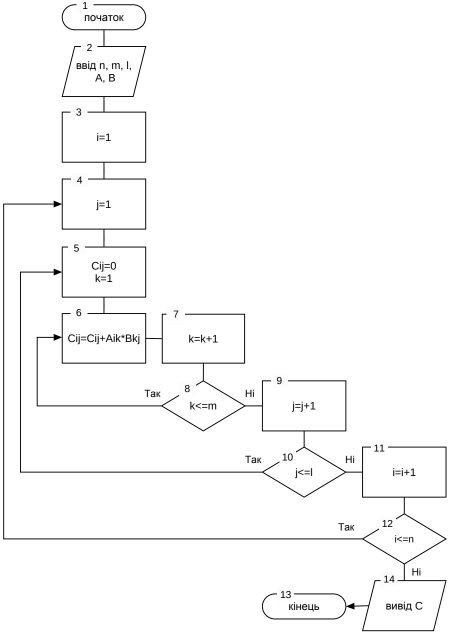
Побудувати алгоритм множення двох прямокутних матриць:

$$A\,(n\times m)\cdot B\,(m\times l) = C\,(n\times l)\,.$$

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1l} \\ b_{21} & b_{22} & \dots & b_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{n2} & \dots & b_{ml} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1l} \\ c_{21} & c_{22} & \dots & c_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nl} \end{pmatrix}$$

$$c_{ij} = \sum_{k=1}^{m} a_{ik}\cdot b_{kj}, \quad 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant l.$$

- A, B, C – матриці – двовимірні масиви;
- n, m, l – розмірності матриць;
- i, j, k – індексні змінні.


1
початок
2
ввід n, m, l,
A, B
3
i=1
4
j=1
5
Cij=0
k=1
6
Cij=Cij+Aik*Bkj
7
k=k+1
8
k<=m
Так
Ні
9
j=j+1
10
j<=l
Так
Ні
11
i=i+1
12
i<=n
Так
Ні
14
вивід C
13
кінець

Приклад 3.

Сортування Неймана

Відсортувати одновимірний масив із n цілих чисел у порядку зростання елементів за допомогою сортування Неймана (“бульбашкове” сортування).

Ідея сортування Неймана

Менші значення виштовхуються **на початок** масиву в той час, як **більші** значення опускаються **в кінець** масиву.

1. Здійснюється декілька проходів по масиву з початку до кінця.
2. При кожному проході відбувається порівняння двох сусідніх елементів:
   - якщо порядок елементів **неправильний**, то вони міняються місцями;
   - якщо порядок елементів **правильний**, то порівнюються два наступні сусідні елементи.
3. Якщо відбулась **хоча би одна** перестановка, то процедура повторюється спочатку.

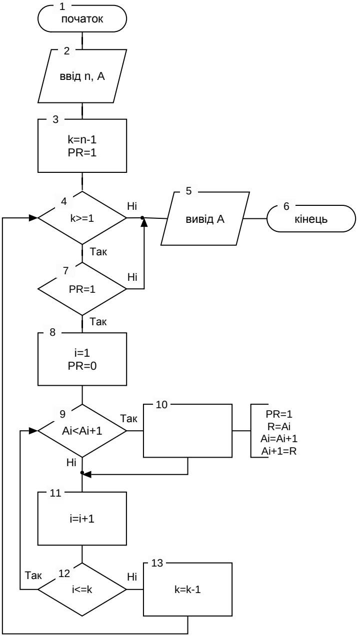
Ідея сортування Неймана

- A – одновимірний масив n елементів;
- i, $1 \leqslant i \leqslant k$ – індексна змінна;
  - $k = n - 1$ – при першому проході;
  - $k = n - 2$ – при другому проході;

  ...
- PR – “прапорець” – індикатор обміну;
  - $PR = 1$ – обмін відбувся;
  - $PR = 0$ – обміну не було.

5	5	5	5	5	4	4	4	3	3
12	4	4	4	4	5	3	3	4	4
4	12	3	3	3	3	5	5	5	5
3	3	12	7	7	7	7	7	7	7
7	7	7	12	12	12	12	12	12	12

1-ий прохід	2-ий прохід	3-ий прохід	4-ий прохід
$k = N - 1$	$k = n - 2$	$k = n - 3$	
$PR = 1$	$PR = 1$	$PR = 1$	$PR = 0$


1
початок
2
ввід n, A
3
k=n-1
PR=1
4
k>=1
Ні
5
вивід A
6
кінець
Так
7
PR=1
Ні
Так
8
i=1
PR=0
9
Ai<Ai+1
Так
10
PR=1
R=Ai
Ai=Ai+1
Ai+1=R
Ні
11
i=i+1
12
i<=k
Так
Ні
13
k=k-1

Приклад 4.

Схема Горнера

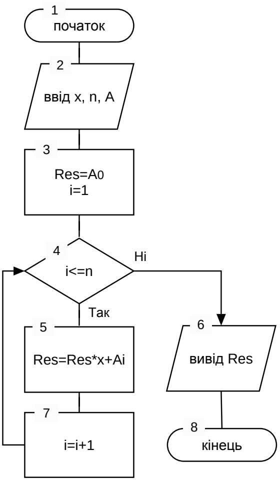
Побудувати алгоритм обчислення значення многочлена

$$P_n(x) = \sum_{k=0}^{n} a_k x^{n-k} = a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$$

за схемою Горнера

$$P_n(x) = \left(\dots \left(\left(\underbrace{a_0}_{\text{ядро}} x + a_1 \right) x + a_2 \right) \dots \right) x + a_n.$$

- n – степінь многочлена;
- A – одновимірний масив коефіцієнтів многочлена;
- i, $0 \leqslant i \leqslant n$ – індексна змінна;
- Res – значення многочлена.


1
початок
2
ввід x, n, A
3
Res=A0
i=1
4
i<=n
Ні
Так
5
Res=Res*x+Ai
6
вивід Res
7
i=i+1
8
кінець

Дякую за увагу!

Далі буде...