Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та структури даних

Лабораторна робота №5

**«Транспонування та множення матриць»**

Виконала:

студентка групи ІО-64

Бровченко А. В.

Перевірив:

Саверченко В. Г.

Київ

2016 р.

**Короткі теоретичні відомості**

***Матриця*** – двомірний масив, елементи котрого розташовуються в рядках та стовпцях, тобто кожний елемент має 2 координати. В дійсності ж не існує двомірних структур в пам’яті, масив записується лінійно, виділяючи лише 2 координати.

*Наприклад,* місця елементів в матриці 3Х3:

|  |  |  |  |
| --- | --- | --- | --- |
|  | a[1,1] | a[1,2] | a[1,3] |
| A = | a[2,1] | a[2,2] | a[2,3] |
|  | a[3,1] | a[3,2] | a[3,3] |

Кожний елемент має 2 координати – i (рядки) та j (стовпці).

*Наприклад*, елемент а[2,1] має координати i=2, j=1.

Елементами матриці можуть бути будь-які дійсні числа.

**Транспонування:**

***Транспонованою матрицею (АТ)*** називається матриця, що виникає з заданої матриці в результаті заміни її рядків на стовпчики та стовпчиків на рядки.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 |  | 1 | 4 | 7 |
| *Наприклад,* **A =** | 4 | 5 | 6 | **АТ** = | 2 | 5 | 8 |
|  | 7 | 8 | 9 |  | 3 | 6 | 9 |

Для транспонування матриці достатньо змінити координати елементів заданої матриці так, щоб координата стовпця стала координатою рядка, а координата рядка – координатою стовпця.

**Множення матриць:**

Нехай задано матриці А та В:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A =** | a[1,1] | a[1,2] | … | a[1,n] | **B =** | b[1,1] | b[1,2] | … | b[1,q] |
| a[2,1] | a[2,2] | … | a[2,n] | b[2,1] | b[2,2] | … | b[2,q] |
| … | … | … | … | … | … | … | … |
| a[m,1] | a[m,2] | … | a[m,n] | b[n,1] | b[n,2] | … | b[n,q] |

Тоді матриця С розміром mXq називається добутком матриць А та В:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| С = | с[1,1] | с[1,2] | … | с[1,q] |
| c[2,1] | c[2,2] | … | c[2,q] |
| … | … | … | … |
| c[m,1] | c[m,2] | … | c[m,q] |

де С[i,j] = (i = 1, 2, …, m; j = 1, 2, …, q).

**Зауваження.** Операція множення справедлива лише тоді і тільки тоді, коли число стовпців в першому співмножнику дорівнює числу рядків в другому.

У випадку коли обидві матриці квадратні одного і того ж порядку, операція множення має місце.

Також існування добутку АВ не гарантує існування добутку ВА та АВ≠ВА.

**Код програми**

**(Python)**

**def** MatrixGen():  
 *"""Генерація матриці"""* print(**'Генеруємо матрицю розмірністю m x n'**)  
 m = int(input(**'m = '**))  
 n = int(input(**'n = '**))  
 Matrix = []  
 **for** i **in** range(0, m):  
 Matrix.append([])  
 **for** j **in** range(0, n):  
 Matrix[i].append(int(input(**'a[{},{}] = '**.format(i, j))))  
 **return** Matrix  
  
  
**def** MatrixOutput(matrix):  
 *"""Вивід матриці"""* **for** i **in** matrix:  
 print(i)  
  
  
**def** MatrixTranspose(Matrix):  
 *"""Транспонування матриці"""  
   
 # Квадратна матриця* **if** len(Matrix) == len(Matrix[0]):  
 TMatrix = Matrix  
 **for** i **in** range(0, len(TMatrix)):  
 **for** j **in** range(0, len(TMatrix[0])):  
 **if** j == i: **break  
 else**:  
 TMatrix[i][j], TMatrix[j][i] = TMatrix[j][i], TMatrix[i][j]  
   
 *# Не квадратна матриця* **else**:  
 TMatrix = []  
 **for** i **in** range(0, len(Matrix[0])):  
 TMatrix.append([])  
 **for** j **in** range(0, len(Matrix)):  
 TMatrix[i].append(0)  
 **for** i **in** range(0, len(TMatrix)):  
 **for** j **in** range(0, len(TMatrix[0])):  
 TMatrix[i][j] = Matrix[j][i]  
 **return** TMatrix  
  
  
**def** MatrixMultiply(Matrix1, Matrix2):  
 *"""Множення матриць Matrix1\*Matrix2"""* **if** len(Matrix1[0]) != len(Matrix2):  
 print(**'Матриці неузгоджені. Неможливо виконати множення.'**)  
 **raise** SystemExit  
  
 Matrix = list([] **for** i **in** range(0, len(Matrix1)))  
 **for** i **in** range(0, len(Matrix1)):  
 **for** j **in** range(0, len(Matrix2[0])):  
 elem = 0  
 **for** k **in** range(0, len(Matrix2)):  
 elem += Matrix1[i][k]\*Matrix2[k][j]  
 Matrix[i].append(elem)  
 **return** Matrix  
  
  
*# Генерація і транспонування*Matrix1 = MatrixGen()  
print(**'Згенерована матриця:'**); MatrixOutput(Matrix1)  
T\_Matrix = MatrixTranspose(Matrix1)  
print(**'Результат транспонування:'**); MatrixOutput(T\_Matrix)  
  
*# На яку матрицю помножити?*option = int(input(**'Починаємо множення матриць\n'  
 'Зробіть свій вибір:\n'  
 '1 -> помножити матрицю саму на себе\n'  
 '2 -> помножити матрицю на транспоновану\n'  
 '3 -> помножити матрицю на нову згенеровану\n'  
 'Ваш вибір: '**))  
**if** option == 1:  
 Matrix2 = Matrix1  
**elif** option == 2:  
 Matrix2 = T\_Matrix  
**elif** option == 3:  
 Matrix2 = MatrixGen()  
 print(**'Нова матриця: '**); MatrixOutput(Matrix2)  
**else**:  
 print(**'Такого варіанту немає.'**)  
 **raise** SystemExit  
  
*# Виконуємо множення*Matrix3 = MatrixMultiply(Matrix1, Matrix2)  
print(**'Результат множення:'**); MatrixOutput(Matrix3)

**Висновок:**

Оптимізацією даної програми є використання функцій. Переваги використання функцій очевидні: по-перше, щоб уникнути зайвих змінних і навіть операцій (звідки і слідує виграш в пам’яті), по-друге, для структурування програми, виділення окремих операцій і блоків.