ITI 1120
Lab #8 - Matrices

Slides by: Romelia Plesa, Sylvia Boyd, Alan Williams, Diana InkPen, Daniel Amyot, Gilbert Arbez, Mohamad Eid

Agenda

- Matrices
  - Example
  - Ex1: Calculating WindChill
  - Ex2: Transposing a Matrix
- If you are ahead…
  - Ex3: Matrix Multiplication

Matrices

- A matrix is a two dimensional rectangular grid of numbers:

$$m = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- The dimensions of the matrix are the numbers of rows and columns (in the above case: row dimension 3, column dimension 3).
- A value within a matrix is referred to by its row and column indices, in that order.
  - Math: number rows and columns from 1, from upper left corner
    - In math notation, $m_{1,2} = 2$
  - For algorithms (and Java), we will use indices starting from 0, to correspond to the array numbering.
    - In algorithm notation, m[0][1] ← 2

3

Matrix element processing

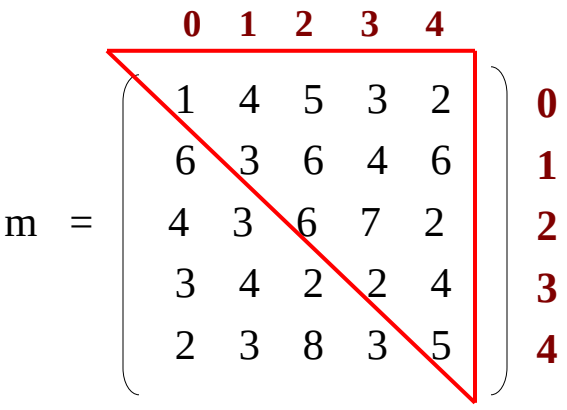
- To visit every element of an array, we had to use a loop.
- To visit every element of a matrix, we need to use a loop inside a loop:
  - Outer loop: go through each row of a matrix
  - Inner loop: go through each column within one row of a matrix.

- Recursion could be used in place of each loop

4

Matrix example

Write an algorithm that finds the sum of the upper triangle of a square matrix (i.e. the diagonal and up).



How do we know if an element of a square matrix is on or above the main diagonal?

line_index <= column_index

5

Matrix example – cont'd

GIVENS:

m	(the matrix)
n	(the number of rows and columns in the matrix)

RESULT:

sum	(the sum of all elements in the upper triangle)

INTERMEDIATES:

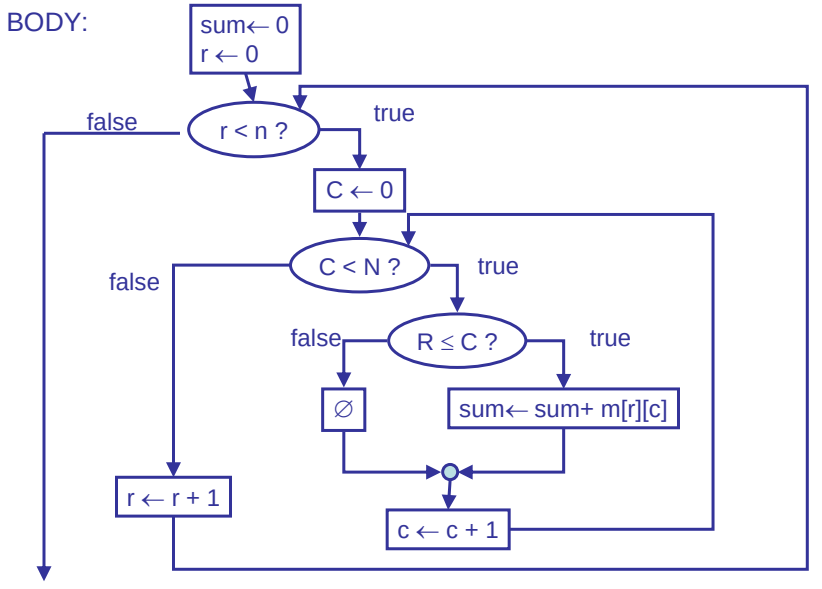
r	(row index in the matrix)
c	column index in the matrix)

HEADER:

sum ← calculateUpperTriangle(m,n)

6

Matrix example – cont'd



7

Matrix Exercise 1: Objective

1. Design an algorithm that will create a matrix of wind chill values for various temperatures and wind speeds.
   - The set of temperatures will be stored in an array
   - The set of wind speeds will be stored in a second array
   - An algorithm is already available that calculates the wind chill for a specific temperature and wind speed.
   - Start with the file Lab8Ex1.doc - you will find already complete the algorithms main and WindChill
2. Implement the algorithm in Java.
   - Start with the file WindChill.java – it provides the complete **`main`** and **`windChill`** methods.

- For those of you who haven't yet experienced an Ottawa winter, you may want to keep the results for future use ☺

8

Given algorithm: Wind Chill calculation

- Yare are given the following algorithm:

  chill ← windChill(temperature, windSpeed)

  that calculates the wind chill for a given temperature in °C, and wind speed in km/hr.

- Restrictions:
  - windSpeed ≥ 5.0 km/hr
  - temperature should be between -50.0°C and +5.0°C.

- More info: http://www.msc.ec.gc.ca/education/windchill/

9

Wind Chill Table

- Design the following algorithm

```
GIVENS:
    tempArray       (an array of temperatures in °C)
    nt              (length of tempArray)
    speedArray      (an array of wind speeds in km/hr)
    ns              (length of speedArray)
RESULTS:
    chillTable      (a matrix of wind chill values, with nt rows
                     and ns columns)
INTERMEDIATES
    (to be determined)
HEADER:
    chillTable ← windChillTable( tempArray, nt, speedArray, ns)
```

10

Implementation in Java

- Get the following files from the lab webpage for lab 8
  - `WindChill.java`
  - `ITI1120.java`
- The file `WindChill.java` already has a `main( )` method as well as an implementation of the method
  `windChill( temperature, windSpeed )`
- The file `WindChill.java` contains the method

```
printTable( temperatureArray, speedArray,
                  chillTable )
```

used to print the matrix.

Sample output

```
Enter a set of temperatures between -50 and +5:
5 0 -5 -10 -15 -20 -25 -30 -35 -40

Enter a set of wind speeds:
5 10 15 20 25 30 35 40

Wind Chill Table

Speed:    5.0  10.0  15.0  20.0  25.0  30.0  35.0  40.0
-------------------------------------------------------
Temp. |
  5.0 |   4.1   2.7   1.7   1.1   0.5   0.1  -0.4  -0.7
  0.0 |  -1.6  -3.3  -4.4  -5.2  -5.9  -6.5  -7.0  -7.4
 -5.0 |  -7.3  -9.3 -10.6 -11.6 -12.3 -13.0 -13.6 -14.1
-10.0 | -12.9 -15.3 -16.7 -17.9 -18.8 -19.5 -20.2 -20.8
-15.0 | -18.6 -21.2 -22.9 -24.2 -25.2 -26.0 -26.8 -27.4
-20.0 | -24.3 -27.2 -29.1 -30.5 -31.6 -32.6 -33.4 -34.1
-25.0 | -30.0 -33.2 -35.2 -36.8 -38.0 -39.1 -40.0 -40.8
-30.0 | -35.6 -39.2 -41.4 -43.1 -44.5 -45.6 -46.6 -47.5
-35.0 | -41.3 -45.1 -47.6 -49.4 -50.9 -52.1 -53.2 -54.2
-40.0 | -47.0 -51.1 -53.7 -55.7 -57.3 -58.7 -59.8 -60.9
```

Lowest value during winters 2003 and 2004 in Ottawa.

Jan. 1994: this value was achieved in Ottawa! (coldest winter since 1948)

Exercice 2: Matrix Transpose

- Write an algorithm that will take a matrix of integers a and transpose the matrix to produce a new matrix a^T. The transpose of the matrix is such that element a_{rc} in the original matrix will be element ${a^T}_{cr}$ in the transposed matrix. The number of rows in a becomes the number of columns in a^T, and the number of columns in a becomes the number of rows in a^T.
  - Start with the Word file Lab8Ex2.doc

- For example:

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad a^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

13

Translate to Java

- Translate your matrix transpose algorithm to Java
  - Start with the file Transpose.java

- Get the following file from labs webpage
  - **MatrixLib.java**

- The file **MatrixLib.java** has the following useful methods (used by main):
  - Method to read a matrix of integers, of specified dimension.

```
public static int[][] readIntMatrix( int numRows, int numCols )
```

  - Method to print a matrix of integers with nice formatting.

```
public static void printMatrix( int[][] matrix )
```

14

Exercise 3: Matrix Multiplication

- Suppose that A is an $m \times n$ matrix and B is an $n \times p$ matrix. The element at row i and column j in A is denoted by a_{ij}.
- Let $C = A \times B$. Then, C will be an $m \times p$ matrix, and for $0 \le i < m$, and $0 \le j < p$,

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj}$$

- Write an algorithm to multiply two given matrices.
  - Start with the Word file Lab8Ex3.doc

15

Translate to Java

- Translate your matrix multiplication algorithm to Java
  - Start with the file Product.java

- The methods from **`MatrixLib.java`** are used by the main method to read and print the matrices.

16