Abou Bakr Belkaïd University - Tlemcen
Faculty of Science
Mathematics Department

TD6 Arrays

Ex 6.1 Exam Notes

We want to write a program that allows us to enter the students' grades and then searches for the maximum, minimum and median grades. To do this:

1. Write a function that asks the user for the number of students N, then reads the N student grades and stores them in an array called NOTES.
2. Write a function that returns the maximum and minimum grade of students' grades.
3. Write a function that calculates and returns the median value of the set of notes.
4. Write the main function to test all the above functions.

Definition of median: The median of a statistical series is a value, noted **Med**, such that the number of values in the series less then **Med** is equal to the number of values greater than **Med**.

- If the total number in the series is odd, the median is the central value of the series.
- If the total number in the series is even, the average of the two central values is usually chosen.

Example :

Let be the series of 16 notes above arranged in ascending order:

2.5	4.5	6	7.25	8	9	10	11	12.75	13	13.5	14.5	15	16	17.5	20

The median of the series is therefore $\frac{11+12.75}{2} = 11.875$.

Ex 6.2 Union and Intersection of two arrays

We have two one-dimensional arrays of integers **T** and **S**. Arrays **T** and **S** are assumed to be already sorted in ascending order and without duplicates. The size of array **T** is ***t*** and the size of array **S** is ***s***.
The variables ***t*** and ***s*** must be less than or equal to the maximum size of **T** and **S** (set here at 100).

1. Write a function **Union** that takes as input two arrays **T** and **S** and their real sizes, and then constructs a **TuS** array of size ***u*** which contains the union of the two arrays **T** and **S**. The **TuS** array should remain sorted by construction.

2. Write a function **Intersection** that takes two arrays **T** and **S** and their real sizes ***t*** and ***s*** as input, and then constructs an array **TnS** of size ***n*** that contains the intersection of the two arrays **T** and **S**. The **TnS** array should remain sorted by construction.

Example :

T :

3	8	17	23	48	56	61	87	98

t = 9.

S :

6	8	23	53	56	76	87

s=7.

TuS :

3	6	8	17	23	48	53	56	61	76	87	98

u=12.

TnS :

8	23	56	87

n=7.

Ex 6.3 Back and Forth

Write a function that transforms a two-dimensional array **TAB** (N rows and M columns) into a linear array **A** (one-dimensional L = N*M). The function assigns the values in the linear array **A** by traversing the two-dimensional array **TAB** back and forth. In other words, the function will traverse the first line of **TAB** from left to right, then the second from right to left, the third from left to right, and so on, alternating the direction of the lines each time.

Example :

TAB

11	8	4
9	5	17
6	1	3
0	14	7

→

A

11	8	4	17	5	9	6	1	3	7	14	0

Ex 6.4 Matrix product

Write a function to multiply a matrix **A** of dimensions N and M with a matrix B of dimensions M and P and that returns the matrix **C** of dimensions N and P:

A(N,M) * **B**(M,P) = **C**(N,P)

Multiplying two matrixes is done by multiplying the components of the two matrixes rows by columns:

$$c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + \cdots + a_{iM} * b_{Mj} = \sum_{k=1}^{M} a_{ik} * b_{kj}$$

Example :

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} * \begin{pmatrix} 7 & 8 \\ 9 & -1 \\ -2 & -3 \end{pmatrix} = \begin{pmatrix} 1*7+2*9-3*2 & 1*8-2*1-3*3 \\ 4*7+5*9-2*6 & 4*8-5*1-6*3 \end{pmatrix} = \begin{pmatrix} 19 & -3 \\ 61 & 9 \end{pmatrix}$$

Ex 6.4 Saddle-points

Write a program to find the elements, in a given matrix A, that are both a maximum on their row and a minimum on their column. These elements are called saddle-points. Display the positions and values of all saddle-points found.

Example :

A[0][0] = 3 is a Saddle-point :

3	2	1
6	5	4
9	8	1

Method: Establish two help matrices MAX and MIN with the same dimensions as A, such as:

$$MAX_{ij} = \begin{cases} 1 & if\ A_{ij}\ is\ a\ maximum\ on\ the\ row \\ 0 & otherwise \end{cases}$$

$$MIN_{ij} = \begin{cases} 1 & if\ A_{ij}\ is\ a\ minimum\ on\ the\ colomn \\ 0 & otherwise \end{cases}$$