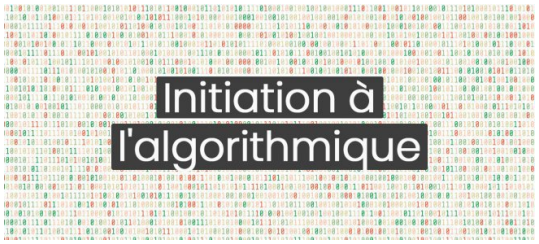
Chapitre 4 : Vecteurs et matrices

ALGORITHMIQUE (ALSD)



Préparé par Dr. BOUCHEBBAH Fatah

UNIVERSITÉ DE BÉJAIA

FACULTÉ DES SCIENCES EXACTES

DÉPARTEMENT D'INFORMATIQUE

Octobre 2022
Version 4

Table des matières

Objectifs 3

I - **Introduction** 4

II - **Définition d'une structure de données** 5

1. Allocation statique 5

2. Allocation dynamique 5

III - **Tableau à une dimension (vecteur)** 6

1. Syntaxe de déclaration d'un vecteur 6

2. Manipulation d'un vecteur 7

IV - **Exercice** 10

V - **Tableau à deux dimensions** 11

1. Syntaxe de déclaration d'une matrice 11

2. Manipulation d'une matrice 12

VI - **Exercice** 14

VII - **Test** 15

Solutions des exercices 16

Références 18

Objectifs

A l'issue de ce chapitre, vous serez capable de :

- Décrire une structure de données statique.
- Illustrer la différence entre un vecteur et une matrice.
- Mettre en œuvre des algorithmes de manipulation des vecteurs et des matrices.

Introduction

Jusqu'à présent, nous avons utilisé des variables de types simples qui peuvent mémoriser une seule valeur à la fois. Imaginons que dans un algorithme, nous avons besoin simultanément de quinze valeurs. Par exemple, quinze notes pour calculer une moyenne d'un étudiant.

Évidemment, la seule solution dont nous disposons à l'état actuel de nos connaissances, consiste à déclarer quinze variables, appelées par exemple : N1, N2, ..., N15. En arrivant au calcul, et après une succession de quinze instructions « LIRE » distinctes, l'unique moyen pour calculer la moyenne est de faire l'opération suivante :

```
1 Moyenne <- (N1+N2+N3+N4+N5+N6+N7+N8+N9+N10+N11+N12+N13+N14+N15)/15 ;
```

Ce qui nous donne un algorithme, de mauvaise qualité, plein de variables et d'opérations longues. C'est pourquoi l'algorithmique nous fournit un outil qui nous permet de rassembler toutes ces variables en une seule, au sein de laquelle chaque valeur sera désignée par un numéro. Ce type de structures de données est appelé un *tableau*.

Définition d'une structure de données

Une structure de données est une manière d'organiser les données pour les traiter plus facilement. En organisant d'une certaine manière les données, on permet un traitement automatique plus efficace et plus rapide de ces dernières p.18 .

Nous distinguos deux types essentiels de structures de données, les structures *statiques* et les structures *dynamiques*. Les mots « statique » et « dynamique » font référence à l'allocation de ces structures en mémoire.

1. Allocation statique

L'allocation statique désigne une réservation inchangeable de l'espace mémoire. Explicitement, la taille à réserver est fixée lors de la déclaration de la variable qui est associée à cet espace mémoire, et ne change pas tout au long de l'exécution du programme utilisant la variable. La taille réservée est considérée gaspillée si elle n'est pas totalement exploitée à un moment donné lors de l'exécution p.18 .

2. Allocation dynamique

L'allocation dynamique décrit une réservation souple qui permet d'allouer et de libérer au besoin un espace mémoire lors de l'exécution d'un programme. De ce fait, l'allocation dynamique garantie une utilisation plus optimale de la mémoire centrale.

Tableau à une dimension (vecteur)

III

Un tableau à une dimension (ou un vecteur) d'ordre (de cardinalité) « n » est un ensemble de « n » valeurs de même type et portant le même nom de variable. Ces valeurs sont identifiées grâce à un nombre appelé indice qui sert à repérer chaque valeur du vecteur séparément.

De ce fait, à chaque fois que nous devons désigner un élément du vecteur, nous faisons figurer le nom du vecteur, suivi de l'indice de l'élément entre deux crochets. La figure ci-dessous représente un vecteur d'entiers :

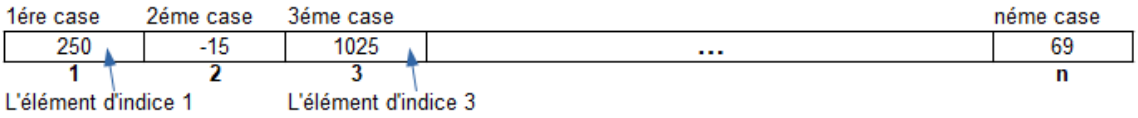


Figure 1. Illustration d'un tableau à une dimension (vecteur) de n éléments.

Une variable vecteur est déclarée en utilisant la réservation statique de la mémoire. Donc, sa taille est fixe du début jusqu'à la fin de l'exécution du programme qui contient cette variable.

1. Syntaxe de déclaration d'un vecteur

Un tableau à une dimension est déclaré en précisant le nombre et le type de valeurs qu'il contiendra.

Syntaxe : Déclaration d'un vecteur

La déclaration d'un vecteur se fait comme suit :

```
1 VAR
2 Nom_tableau : TABLEAU [indice_minimum .. indice_maximum] DE type_des_éléments ;
```

Ou en langage C :

```
1
2 type_des_éléments : Nom_tableau [nombre_d'éléments];
```

Exemple : Déclaration d'une variable vecteur

```
1 Tab : TABLEAU [1 .. 100] D'Entier ;
2 Note : TABLEAU [1 .. 10] DE Réel ;
```

Pour la variable *Tab* déclarée ci-dessus :

- Le nom de la variable est *Tab*, il est *fixe*.
- La taille de la variable est *100*. Elle est *fixe* car la réservation utilisée est statique.
- Le type des valeurs que contiendra la variable est *Entier*.

Complément : Plus de détails

- Lorsque nous déclarons un vecteur, nous déclarons aussi de façon implicite toutes les variables indicées qui le constituent.
- En règle générale, l'indice minimum vaut 1. Mais dans certains langage de programmation, on utilise l'indice 0, comme indice minimum. Lorsque l'indice minimum est 1, la taille du vecteur (nombre d'éléments) est égale à la valeur de l'indice maximum.
- Le type de l'indice doit être un type ordinal (*i.e.* Entier mais jamais Réel).

2. Manipulation d'un vecteur

Les éléments d'un vecteur sont des variables indicées qui sont utilisées exactement comme n'importe quelles autres variables classiques p.18 . Autrement dit :

- Elles peuvent faire l'objet d'une affectation.
- Elles peuvent figurer dans une expression arithmétique ou dans une comparaison.
- Elles peuvent être affichées et saisies.

Nous accédons aux éléments d'un tableau en utilisant la syntaxe suivante : *Nom_tableau [indice]*, c'est la même syntaxe que nous utilisons en langage C.

Exemple : Illustrations

Si *Tab1* est un tableau de 10 entiers alors :

- L'instruction Tab1 [2] <- 5 permet de mettre la valeur 5 dans la deuxième case du vecteur.
- En considérant le cas où *A* est une variable de type entier, l'instruction A <- Tab1 [2] permet de mettre la valeur de la deuxième case du vecteur dans *A* (*i.e.* A <- 5).
- LIRE (Tab1 [1]) permet de mettre l'entier saisi par l'utilisateur dans la première case du vecteur.
- ÉCRIRE (Tab1 [2]) permet d'afficher sur l'écran la valeur de la deuxième case du tableau.

Remarque : Gardez en tête !

- Il ne faut pas confondre l'indice d'un élément et la valeur de cet élément. En d'autres termes, il ne faut pas confondre « i » avec « Tab [i] ; ».
- De même qu'une variable possède toujours une valeur (qui peut être indéterminée), un tableau peut très bien posséder des éléments dont la valeur est indéterminée. Donc avant d'utiliser un tableau, il est donc nécessaire de l'initialiser.

Exemple : Remplir et afficher un vecteur

Écrire un algorithme permettant de remplir un tableau de 10 cases d'entiers, ensuite d'afficher le contenu des cases.

```
ALGORITHME Lecture_Affichage ;
CONST N = 10 ;
VAR
    Tab : TABLEAU [1..N] d'Entier;
    i : Entier;
DÉBUT
   POUR i ALLANT DE 1 A N FAIRE
      LIRE (Tab[i]) ;
  FIN POUR ;
  POUR i ALLANT DE 1 A N FAIRE
    ÉCRIRE (Tab[i]) ;
  FIN POUR ;
FIN.
```

Exemple : Additionner les éléments de deux vecteurs

Nous disposons de deux vecteurs T1 et T2 et nous voulons réaliser la somme élément par élément des deux vecteurs. Ça consiste à additionner les éléments de même indice et les mémoriser dans un vecteur T3.

```
ALGORITHME Add_Tab ;
CONST M = 100;
VAR
 T1,T2,T3 : TABLEAU [1..M] d'Entier ;
 i, N : Entier;
DÉBUT
 RÉPÉTER
  ÉCRIRE (' Donnez la taille des vecteurs');
  LIRE (N);
 JUSQU'A (N <= M) ET (N>0);
 POUR i ALLANT DE  1 à N FAIRE
  LIRE(T1[i]) ;
 FIN POUR;
 POUR i ALLANT DE 1 à N FAIRE
  LIRE(T2[i]) ;
 FIN POUR ;
 POUR i ALLANT DE 1 à N FAIRE
  T3[i] <- T1[i] + T2[i] ;
 FIN POUR ;
 POUR i ALLANT DE 1 à N FAIRE
 ÉCRIRE (T3[i]) ;
 FIN POUR;
FIN.
```

Exemple : Recherche séquentielle dans un vecteur

Écrire un algorithme qui recherche une valeur, lue à partir du clavier, dans un vecteur. L'algorithme doit renvoyer la position de la valeur recherchée si elle existe.

```
ALGORITHME Recherche;
CONT M = 20;
VAR
 V : TABLEAU[1..M] D'Entier ;
 D, i: Entier ;
 Trouve : Booléen ;
DÉBUT
```

```
(* remplir le vecteur V *)
POUR i ALLANT DE  1 à M FAIRE
 LIRE(V[i]) ;
FIN POUR;
(* lire la valeur recherchée *)
ÉCRIRE ('Veuillez saisir la valeur recherchée') ;
LIRE(D) ;
(* vérifier si la valeur rechérchée existe dans le vecteur *)
i <- 1 ;
Trouve <- Faux ;
TANT QUE (i <= M) et (NON Trouve) FAIRE
 SI (V[i] = D) ALORS
  Trouve <- Vrai ;
 SINON
  i <- i + 1 ;
 FIN SI;
FIN TANT QUE ;
(* afficher la position de la valeur recherchée si elle existe *)
SI (Trouve = Vrai) ALORS
 ÉCRIRE ('La position de la valeur recherchée est : ', i) ;
SINON
 ÉCRIRE ('La valeur recherchée n'existe pas dans le tableau') ;
FIN SI;
FIN.
```

Exercice

IV

[solution n°1 p.16]

Lesquelles des affirmations suivantes sont correctes ?

- ☐ Un vecteur est une variable qui peut stocker plusieurs valeurs simultanément.
- ☐ Un vecteur est une constante qui peut stocker plusieurs valeurs simultanément.
- ☐ On déclare un vecteur dans le corps de l'algorithme.
- ☐ On ne peut pas manipuler les cases d'un vecteur avec les instructions LIRE et ÉCRIRE.
- ☐ Dans la l'instruction T [5] <- 6 ; la case ayant l'indice 5 recevra la valeur 6.

Tableau à deux dimensions

V

Nous appelons tableau à deux dimensions (ou matrice) d'ordre « (m × n) » un ensemble de « m × n » éléments rangés dans les cases disposés en « m » lignes et « n » colonnes. Dans ce type de tableaux les valeurs ne sont pas repérées par un seul indice comme dans les vecteurs, mais par deux indices (coordonnées). Le premier indice sert à représenter les lignes et le second indice sert à représenter les colonnes p.18 . La figure ci-dessous représente un tableau à deux dimensions :

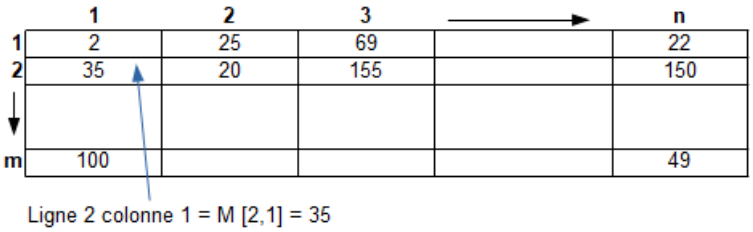


Figure 2. Illustration d'une matrice.

1. Syntaxe de déclaration d'une matrice

Un tableau à deux dimensions est déclaré en précisant le nombre de ses lignes et ses colonnes, et le type des valeurs qu'il contiendra.

Syntaxe : Déclaration d'une matrice

La déclaration d'une matrice se fait de la manière citée ci-dessous :

```
VAR
Nom_matrice : TABLEAU [inf_ligne .. max_ligne, inf_colonne .. max_colonne] DE
type_des_éléments ;
```

Ou en langage C :

```
Type_des_éléments : Nom_matrice [nombre_d'éléments_dimension1]
[nombre_d'éléments_dimension2];
```

Exemple : Déclaration d'une matrice

```
Achat : TABLEAU [1 ..100, 1 .. 100] d'Entier ;
Vente : TABLEAU [1 .. 10, 1 .. 47] de Réel ;
```

Pour la variable *Achat* déclarée ci-dessus :

- Le nom de la variable est *Achat*, il est *fixe*.
- La taille de la variable est *(100 × 100)*. Elle est *fixe* car la réservation utilisée est statique.
- Le type des valeurs que contiendra la variable est *Entier*.

Complément : Plus de détails

- A la déclaration d'une matrice, il faut ajouter la déclaration de deux indices qui doivent être d'un type ordinal.
- Tous les éléments de la matrice doivent être de même type.
- Chaque élément de la matrice est repéré par le nom de la matrice et les numéros des indices de ligne et de colonnes, on le note M [i, j].
- Lorsque m=n, alors la matrice est dite carrée.
- Lorsque la matrice est carrée et si i=j, alors l'élément se trouve sur la diagonale principale de la matrice.
- La saisie et l'affichage des éléments de la matrice sont effectuées par les instructions de lecture et d'écriture.

2. Manipulation d'une matrice

Pour insérer (ou afficher) des données dans (à partir) une (d'une) matrice à (n × m) cases il faut utiliser une boucle qui parcoure les lignes et une autre imbriquée qui parcoure les colonnes, puisqu'une case d'une matrice est au croisement d'une ligne et d'une colonne p.18 .

Exemple : Remplir et afficher les éléments d'une matrice

Pour remplir une matrice *Mate* de (3× 4) cases (12 cases) :

```
1 ALGORITHME Lectur_Mat ;
2 CONST
3     N = 3 ;
4     M = 4 ;
5 VAR
6     Mate : TABLEAU [1 .. N, 1 .. M] d'Entier ;
7     i,j: Entier;
8 DÉBUT
9    POUR i ALLANT DE 1 A N FAIRE
10      POUR j ALLANT DE 1 A M FAIRE
11        LIRE (Mate [i , j]) ;
12      FIN POUR ;
13   FIN POUR ;
14 FIN.
```

Pour afficher le contenu des cases d'une matrice, on utilise la même structure de lecture :

```
1 ALGORITHME Ecriture_Mat ;
2 VAR
3     Mate : TABLEAU [1 .. 3, 1 .. 4] d'Entier ;
4     i,j: Entier;
5 DÉBUT
6    POUR i ALLANT DE 1 A N FAIRE
```

```
      POUR j ALLANT DE 1 A M FAIRE
        ÉCRIRE (Mate [i , j]) ;
      FIN POUR ;
   FIN POUR ;
FIN.
```

Exemple : Additionner les éléments de deux matrices

On dispose de deux matrices M1 et M2 et on veut réaliser la somme élément par élément des deux matrices. Ça consiste à additionner les éléments de mêmes indices et les mémoriser dans une matrice M3.

```
ALGORITHLE ADD_Mat;
CONST
 NL = 10;
 NC = 20 ;
VAR
 Mat1,Mat2 : TABLEAU [1..NL,1..NC] d'Entier ;
 i,j : Entier ;
DÉBUT
 (* remplir la matrice Mat1 *)
 POUR i ALLANT DE 1 à NL FAIRE
  POUR j ALLANT DE 1 à NC FAIRE
   LIRE(Mat1 [i, j]);
  FIN POUR ;
 FIN POUR ;
 (* remplir la matrice Mat2 *)
 POUR i ALLANT DE 1 à NL FAIRE
  POUR j ALLANT DE 1 à NC FAIRE
   LIRE(Mat2 [i, j]);
  FIN POUR ;
 FIN POUR ;
 (* additionner des deux matrices *)
 POUR i ALLANT DE 1 à NL FAIRE
  POUR j ALLANT DE 1 à NC FAIRE
   Mat3[i, j]<- Mat1[i,j] + Mat2[i,j] ;
  FIN POUR ;
 FIN POUR ;
(* afficher de la matrice résultante Mat3 *)
 POUR i ALLANT DE 1 à NL FAIRE
  POUR j ALLANT DE 1 à NC FAIRE
   ÉCRIRE (Mat3 [i, j]);
  FIN POUR ;
 FIN POUR ;
FIN.
```

Exercice

VI

[solution n°2 p.16]

Lesquelles des affirmations suivantes sont correctes ?

- ☐ Une matrice est une structure de données dynamique.
- ☐ Une matrice est une structure de données statique.
- ☐ Une matrice est composé de lignes et de colonnes.
- ☐ Nous avons besoin de deux boucles pour lire et afficher une matrice.

Test

VII

Exercice

[solution n°3 p.16]

En algorithmique, dans un vecteur V de N cases, la première case a pour indice et le contenu de cette case est accessible en utilisant l'instruction .

Exercice

[solution n°4 p.16]

On considère deux tableaux T1 et T2. Peut-on copier le contenu de T2 dans T1 sans perdre d'information?

- O Non, impossible.
- O Élément par élément à l'aide d'une boucle si la taille de T1 est >= à la taille de T2 ?
- O Élément par élément à l'aide d'une boucle si la taille de T1 est <= à la taille de T2 ?
- O Oui, on peut copier les données de T2 dans T1 sans aucune restriction.

Exercice

[solution n°5 p.17]

Quelle(s) déclaration(s) correspond(ent) à une matrice de N lignes et M colonnes?

- ☐ Tab : TABLEAU [1 .. M, 1 .. N] d'entiers ;
- ☐ Tab : TABLEAU [1 .. N, 1 .. M] de Réel ;
- ☐ Tab : TABLEAU [1 .. N, 1 .. M] d'Entier ;
- ☐ M : TABLEAU [1 .. N, 1 .. M] de caractères ;
- ☐ M : TABLEAU [1 .. N, 1 .. 100] d'Entier ;

Solutions des exercices

> **Solution** n°1

Exercice p. 10

Lesquelles des affirmations suivantes sont correctes ?

- [x] Un vecteur est une variable qui peut stocker plusieurs valeurs simultanément.
- [ ] Un vecteur est une constante qui peut stocker plusieurs valeurs simultanément.
- [ ] On déclare un vecteur dans le corps de l'algorithme.
- [ ] On ne peut pas manipuler les cases d'un vecteur avec les instructions LIRE et ÉCRIRE.
- [x] Dans la l'instruction T [5] <- 6 ; la case ayant l'indice 5 recevra la valeur 6.

> **Solution** n°2

Exercice p. 14

Lesquelles des affirmations suivantes sont correctes ?

- [ ] Une matrice est une structure de données dynamique.
- [x] Une matrice est une structure de données statique.
- [x] Une matrice est composé de lignes et de colonnes.
- [x] Nous avons besoin de deux boucles pour lire et afficher une matrice.

> **Solution** n°3

Exercice p. 15

En algorithmique, dans un vecteur V de N cases, la première case a pour indice `1` et le contenu de cette case est accessible en utilisant l'instruction `V [1]`.

> **Solution** n°4

Exercice p. 15

On considère deux tableaux T1 et T2. Peut-on copier le contenu de T2 dans T1 sans perdre d'information?

- ○ Non, impossible.
- ◉ Élément par élément à l'aide d'une boucle si la taille de T1 est >= à la taille de T2 ?
- ○ Élément par élément à l'aide d'une boucle si la taille de T1 est <= à la taille de T2 ?
- ○ Oui, on peut copier les données de T2 dans T1 sans aucune restriction.

> **Solution** n°5

Exercice p. 15

Quelle(s) déclaration(s) correspond(ent) à une matrice de N lignes et M colonnes?

- ☐ Tab : TABLEAU [1 .. M, 1 .. N] d'entiers ;
- ☑ Tab : TABLEAU [1 .. N, 1 .. M] de Réel ;
- ☑ Tab : TABLEAU [1 .. N, 1 .. M] d'Entier ;
- ☑ M : TABLEAU [1 .. N, 1 .. M] de caractères ;
- ☐ M : TABLEAU [1 .. N, 1 .. 100] d'Entier ;

Références

1

L. Baba-Hamed, S. Hocine. Algorithmique et structures de données : Cours et exercices avec solutions. Office des Publications Universitaires, 2006

2

M. C. Belaid. Algorithmique et programmation en Pascal, cours, exercices et travaux pratique avec corrigés. Edition Les Pages Bleues, 2004.

3

M. C. Belaid. Algorithme et programmation en Pascal. Edition les pages bleus, 2006.

4

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Algorithmique : Cours avec 957 exercices et 158 problèmes. Édition DUNOD, 3ème édition, 2010.

5

J. Courtin. Initiation à l'algorithmique et aux structures de données. Edition DUNOD, 1998.

6

M. Divay. Algorithmes et structures de données génériques. Edition Dunod, 2004.

7

L. Goldschlager and A. Lister. Informatique et algorithmique. InterEditions. 1986.

8

https://9alami.info/cours-informatique/liste-des-modules/lecon1-notion-dalgorithme (Consulté le 09/03 /2021)

9

https://www.youtube.com/playlist?list=PL2aehqZh72Lumvy4tSekr6Rzcgwn15MLI (Consulté le 05/04 /2021)