

# FICHE DE RÉVISION :

## - ALGORITHMIQUE -

### TABLEAUX À UNE DIMENSION



#### STRUCTURE DE DONNÉES :

→ manière d'organiser les données pour les traiter plus facilement

Elle est caractérisée par :

- le nombre d'informations qu'elle peut contenir
- le type de ces informations
- le mode d'accès à ces informations

#### MODES D'ACCÈS

- **Séquentiel** : on lit les éléments de la structure de données dans l'ordre.
- **Direct par position** : accéder directement à un élément via sa position (= son indice).
- **Direct par nom** : accéder directement à un élément via son nom.
- **Aléatoire** : accéder à une information quelconque.

## TABLEAUX À UNE DIMENSION

#### DÉFINITION :

Structure de données permettant de **stocker un nombre fini, non nul, connu à l'avance, d'éléments tous de même type.** ❤️

#### CARACTÉRISTIQUES

- **Taille du tableau** : constante, implicite ou explicite, non modifiable.
- **Aucune restriction** sur le type d'élément ❤️
- **Mode d'accès** : direct par position (indice de 1 à n).
- **Débordement** = erreur produite si on tente d'accéder à un élément d'indice  $i$  supérieure à la taille du tableau  $n$ .
- **Intérêt** : stockage grand nb de variables d'un même type avec une seule déclaration + accès simplifié via indice

#### SYNTAXE

##### VARIABLE

tab : **TABLEAU DE** n Typ

- **tab** = identificateur qui repère la variable de type tableau
- **TABLEAU DE** : mot réservé définissant un tableau
- **n** = taille (entier > 0) : cste implicite ou explicite
- **Typ** = type des éléments (NUMERIQUE, CHAINE, BOOLEEN...)
- **tab [i]** : élément de tab situé à la position  $i$  avec  $1 \leq i \leq n$

#### DÉCLARATION

→ se fait dans la section de déclaration des variables :

##### VARIABLE

tab : **TABLEAU DE** <valeur> <type>

avec <valeur> : taille  $n$  du tableau

<type> : type des éléments du tableau

#### ACCÈS ET MODIFICATIONS DES ÉLÉMENTS

- **tab[i]** : élément à l'indice  $i$  dans le tableau tab
- **tab[i] ← x** : affecter la valeur  $x$
- **ECRIRE(tab[i])** : afficher l'élément

## ALGORITHMES SUR LES TABLEAUX À UNE DIMENSION

Pour manipuler un tableau, on utilise des algorithmes de parcours permettant de **traiter chaque élément une fois.**

On distingue 2 types de parcours : **exhaustif**, où l'on traite tous les éléments du tableau ; **conditionnel**, où l'on ne traite que les éléments qui vérifient une condition (expression booléenne).

Le choix du type de parcours dépend de l'objectif de l'algorithme.

#### PARCOURS EXHAUSTIF

→ On effectue un traitement **sur tous les éléments** contenus dans le tableau.

On utilise une **itération à bornes fixes (boucle POUR)** car le nombre d'éléments est connu à l'avance grâce à la taille du tableau.

```
SSALGO Parcours exhaustif(tab : TABLEAU DE Taille NUMERIQUE PAR VALEUR/REFERENCE)
// parcours exhaustif d'un tableau tab de Taille nombres
// Taille est une constante globale
// tab est passé par valeur si on ne souhaite pas qu'il soit modifié, par référence sinon
VARIABLE
  i: NUMERIQUE // indice d'itération
DEBUT
  POUR i DE 1 A Taille FAIRE
    Traitement // on applique les instructions à tab[i]
FPOUR
FINSA
```



## PARCOURS CONDITIONNEL

→ S'arrête dès qu'une **condition est vérifiée** par `tab[i]`. On utilise une itération conditionnelle (**TANTQUE**)

```

SSALGO Parcoursconditionnel(tab : TABLEAU DE Taille NUMERIQUE PAR VALEUR/REFERENCE)
// parcours conditionnel d'un tableau tab de Taille nombres
// Taille est une constante globale
// tab est passé par valeur si on ne souhaite pas qu'il soit modifié, par référence sinon
VARIABLE
  i: NUMERIQUE // indice d'itération
DEBUT
  i ← 1
TANTQUE ((i <= Taille) ET condition) FAIRE
  Traitement /* on applique les instructions à tab[i], au moins une instruction permet de modifier l'évaluation de la
condition */
  i ← i+1 // on passe à l'indice suivant
FTQUE
FINSA

```

## TABLEAUX TRIÉS

Trier un tableau = ordonner ses éléments selon une relation d'ordre (ex. croissant).

L'objectif est de le faire sans utiliser de tableau supplémentaire.

Il existe plusieurs méthodes de tri (tri à bulles, tri par sélection, etc.) qui se distinguent par leur complexité, c'est-à-dire le temps d'exécution ou la mémoire utilisée par l'algorithme.

### TRI À BULLES

→ méthode de tri simple qui consiste à **comparer les éléments consécutifs du tableau tab et à les permuter lorsqu'ils sont mal triés.**

À chaque parcours complet, le **plus grand élément «remonte» à la fin du tableau**, à sa position définitive.

L'opération est répétée jusqu'à ce que tout le tableau soit trié.

```

SSALGO Tribulles(tab : TABLEAU DE Taille NUMERIQUE PAR REFERENCE)
// tri à bulles d'un tableau tab de Taille éléments de type NUMERIQUE
/* tab est une donnée/résultat : tab est fourni non trié en entrée du sous-algorithme et est trié dans le sous-algorithme pour
être rendu trié, c'est pourquoi il est passé par référence */
VARIABLE
  i, j: NUMERIQUE // indices d'itération
DEBUT
  // on a deux boucles imbriquées
POUR i DE (Taille-1) A 1 (PAS - 1) FAIRE // itération qui va placer le plus grand élément restant à la position i+1
  POUR j DE 1 A i FAIRE // itération parcourant les éléments restant à trier
    SI (tab[j+1] < tab[j])
      ALORS Echange(tab[j+1], tab[j]) // sous-algorithme de Leçon 4 : Exemples
    FSI
  FPOUR
FPOUR
FINSA

```

## GLOSSAIRE

- **structure de données** : manière d'organiser de l'information
- **type structuré** : type défini permettant de représenter des structures de données (tableaux, articles, listes, arbres...)
- **parcours de structure de données** : algorithme permettant de passer une fois et une seule par chaque élément, pour appliquer un traitement.
- **mode d'accès** : façon d'accéder à un élément de la structure de données.
- **tableau à une dimension** : structure de données permettant de stocker un nombre fini, non nul, connu à l'avance, d'informations toutes de même type ; l'accès à une information direct par position.
- **élément / case** : information contenue dans un tableau.