



ALGO2

Département Mathématiques et Informatique

SMI / S3

Achtaich Khadija

ALGO2

Objectives



Achtaich Khadija

ALGO 2



Plan du cours



01 Chapitre 1
Introduction et rappels

02 Chapitre 2
Les tableaux

03 Chapitre 3
Les procédures et les fonctions

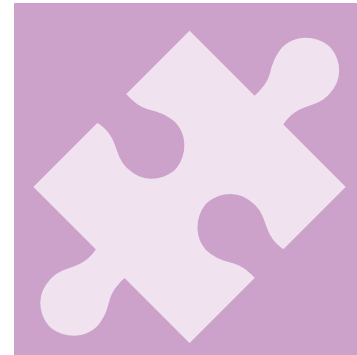
04 Chapitre 4
La récursivité

05 Chapitre 5
Les enregistrements et les fichiers

06 Chapitre 6
La complexité

ALGO 2

CHAPITRE 1



Introduction et rappels

Achtaich Khadija

Introduction

Qu'est-ce qu'un algorithme?

Est une suite d'instructions écrite en langage d'algorithme qui résout un problème et qui peuvent être programmé par n'importe quel langage.

Une suite d'instructions serait :

1. Se lever
2. Se laver le visage
3. Prendre le petit déjeuner
4. S'habiller

Achtaich Khadija



Pourquoi faire des algorithmes ?

la rédaction des algorithmes permet plusieurs choses :

- d'être compréhensible par tout *informaticien* même s'il ne connaît pas le langage du programme
- de vérifier la complexité du programme et donc de l'optimiser
- de faire ressortir de manière compréhensible les cas d'utilisations

Achtaich Khadija



Définition

Algorithme:

Définition : Un algorithme est un ensemble d'opérations de calcul élémentaires, organisé selon des règles précises dans le but de résoudre un problème donné. Pour chaque donnée du problème, l'algorithme retourne une réponse après un nombre fini d'opérations(+, -, /, <, >, ...).

Achtaich Khadija



Méthodologie

La résolution d'un problème est caractérisé par 4 étapes :

- Comprendre la nature du problème posé
- Préciser les données fournies (Entrées)
- Préciser les résultats que l'on désire obtenir (Sorties)
- Déterminer le processus de transformation des données en résultats.

Achtaich Khadija



Notions de Base

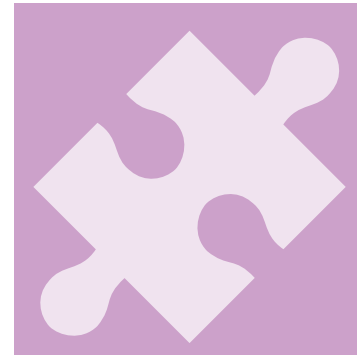
- Comment faire des algorithmes?
- Les variables
- Le type de la variable
- Les instructions
- Syntaxe générale de l'algorithme
- Les structures de contrôle
- Structure répétitive

Achtaich Khadija



ALGO 2

CHAPITRE 2



Les tableaux

Achtaich Khadija

Les tableaux

Supposons que nous avons besoin de calculer la moyenne de 12 notes d'un étudiant.

la seule solution dont nous disposons consiste à déclarer douze variables, appelées par exemple: X1, X2, X3,...X12.

La première étape est de lire les valeurs de toute ces variables une par une, ce qui nous fait douze instructions de lecture et après calculer la moyenne par l'instruction:

$$\text{Moy} \leftarrow (X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12)/12$$

Achtaich Khadija



Les tableaux(suite)

C'est pourquoi l'algorithmique (la programmation) nous permet de rassembler toutes ces variables en une seule, au sein de laquelle chaque valeur sera désignée par un numéro.

Achtaich Khadija



Les tableaux(suite)

Notation et utilisation algorithmique

- Dans notre exemple, nous créerons donc un tableau appelé Notes.
- **Tableau** Notes(12) : **Entier**
- On peut créer des tableaux contenant des variables de tous types : tableaux de numériques, tableaux de caractères, tableaux de booléens, tableaux de tout ce qui existe dans un langage donné comme type de variables.
- L'énorme avantage des tableaux, c'est qu'on va pouvoir les traiter en faisant des boucles.

Achtaich Khadija



Lecture d'un tableau

Notation et utilisation algorithmique

Tableau $T(10)$: Entier
Variables i : Entier

```
Pour  $i \leftarrow 0$  à 9  
    Ecrire ("Entrez l'élément n°",  $i+1$ )  
    Lire( $T(i)$ )  
Fpour
```

Achtaich Khadija



Affichage d'un tableau

Notation et utilisation algorithmique

```
Ecrire (" Voilà votre tableau:")  
Pour i ← 0 à 9  
    Ecrire(T(i))  
Fpour
```

Achtaich Khadija



Les tableaux(suite)

Notation et utilisation algorithmique

```
Tableau Note(12) : Entier
Variables i, Som : Entier
Variable Moy : Réel
Pour i ← 0 à 11
    Ecrire ("Entrez la note n°", i+1)
    Lire( Note(i))
FPour
Som ← 0
Pour i ← 0 à 11
    Som = Som + Note(i)
Fpour
Moy = Som / 12
```

Achtaich Khadija



Les tableaux(suite)

Remarque générale : l'indice qui sert à désigner les éléments d'un tableau peut être exprimé directement comme un nombre en clair, mais il peut être aussi une variable, ou une expression calculée.

La valeur d'un indice doit toujours :

- **être égale au moins à 0** (dans quelques rares langages, le premier élément d'un tableau porte l'indice 1). Mais nous avons choisi ici de commencer la numérotation des indices à zéro, comme c'est le cas en langage C.
- **être un nombre entier.** Quel que soit le langage.
- **être inférieure ou égale au nombre d'éléments du tableau (moins 1, si l'on commence la numérotation à zéro).**

Achtaich Khadija



Les tableaux(suite)

EXEMPLE 1:

Que produit l'algorithme suivant ?

Tableau A(7) :Entier

Variable i, x, y :Entier

Début

$x \leftarrow A(0)$

$y \leftarrow A(0)$

Pour i \leftarrow 0 à 6

Si $x < A(i)$ **alors**

$x \leftarrow A(i)$

Finsi

Si $y > A(i)$ **alors**

$y \leftarrow A(i)$

Finsi

FinPour

Ecrire x

Ecrire y

Fin

Achtaich Khadija



Les tableaux(suite)

EXEMPLE 2:

Ecrire l'algorithme qui affiche l'indice du minimum d'un tableau de 100 entiers.

Solution :

Tableau A(100) : Entier

Variable i, indiceMin : Entier

Début

indiceMin ← 0

pour i ← 0 **à** 99 **faire**

si A(i) < A(indiceMin) **alors**

 indiceMin ← i

Fsi

Finpour

Ecrire indiceMin

Fin

Achtaich Khadija



Tableaux multidimensionnels

Dans un tableau simple une valeur est repérée par une seule coordonnée: i
par exemple

- Dans un tableau multidimensionnel elle est repérée par deux coordonnées ou plus.
- les indices des tableaux ne sont que des coordonnées logiques, pointant sur des adresses de mémoire vive, ainsi un tableau peut être de dimension n .

Achtaich Khadija



Tableaux à deux dimensions

L'informatique nous offre la possibilité de déclarer des tableaux dans lesquels les valeurs ne sont pas repérées par une seule, mais par deux coordonnées.

Un tel tableau se déclare de la manière suivante :

Tableau nomdutableau (taille1, taille2): type

Exemple:

Tableau T (10, 12): entier

Achtaich Khadija



Tableaux à deux dimensions

Ainsi dans le parcours on utilise deux boucles imbriquées:

- La première pointe les lignes (ligne par ligne)
- La deuxième pointe les cases d'une ligne (case par case)

- **Remarque :** la notion de lignes et de colonnes est utilisée pour comprendre mais n'a aucun sens puisqu'il s'agit de coordonnées pouvant être de dimension n .

Achtaich Khadija



Tableaux à deux dimensions

Exemple : Écrivez un algorithme remplissant un tableau de 6 sur 13, avec des zéros.

Réponse:

Tableau $T(6, 13)$: Entier

Début

Pour $i \leftarrow 0$ à 5

 Pour $j \leftarrow 0$ à 12

$T(i, j) \leftarrow 0$

 FinPour j

FinPour i

Fin

Achtaich Khadija



Tableaux à deux dimensions

Exercice 1: Quel résultat produira cet algorithme ?

Tableau $T(4, 2)$: Entier

Variables k, m : Entier

Début

Pour $k \leftarrow 0$ à 3

 Pour $m \leftarrow 0$ à 1

$T(k, m) \leftarrow k + m$

 FinPour m

FinPour k

Pour $k \leftarrow 0$ à 3

 Pour $m \leftarrow 0$ à 1

 Ecrire($T(k, m)$)

 FinPour m

FinPour k

Fin

Achtaich Khadija



Tableaux à deux dimensions

Solution Exercice 1:

Cet algorithme remplit un tableau de la manière suivante:

$$T(0, 0) = 0$$

$$T(0, 1) = 1$$

$$T(1, 0) = 1$$

$$T(1, 1) = 2$$

$$T(2, 0) = 2$$

$$T(2, 1) = 3$$

$$T(3, 0) = 3$$

$$T(3, 1) = 4$$

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

Achtaich Khadija



Tableaux à deux dimensions

Exercice 2: Quel résultat produira cet algorithme ?

```
Tableau X(2, 3) :Entier
Variables i, j, val : Entier
Début
  Val ← 1
  Pour i ← 0 à 1
    Pour j ← 0 à 2
      X(i, j) ← Val
      Val ← Val + 1
    FinPour j
  FinPour i
  Pour j ← 0 à 2
    Pour i ← 0 à 1
      Ecrire (X(i, j))
    FinPour i
  FinPour j
Fin
```

Achtaich Khadija



Tableaux à deux dimensions

Solution Ex2:

Cet algorithme remplit un tableau de la manière suivante:

$$X(0, 0) = 1$$

$$X(0, 1) = 2$$

$$X(0, 2) = 3$$

$$X(1, 0) = 4$$

$$X(1, 1) = 5$$

$$X(1, 2) = 6$$

Il écrit ensuite ces valeurs à l'écran, dans cet ordre.

$$X(0, 0) = 1$$

$$X(1, 0) = 4$$

$$X(0, 1) = 2$$

$$X(1, 1) = 5$$

$$X(0, 2) = 3$$

$$X(1, 2) = 6$$

Achtaich Khadija



Tableaux à deux dimensions

Exercice 3 :

Soit un tableau T à deux dimensions $(13, 9)$ préalablement rempli de valeurs numériques entières.

Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau

Achtaich Khadija



Tableaux à deux dimensions

Solution Ex3:

Variables i, j, iMax, jMax: entier

Tableau T(13, 9): réel

Debut

iMax \leftarrow 0

jMax \leftarrow 0

Pour i \leftarrow 0 à 12

 Pour j \leftarrow 0 à 8

 Si $T(i,j) > T(iMax,jMax)$ Alors

 iMax \leftarrow i

 jMax \leftarrow j

 FinSi

 FinPour j

FinPour i

Ecrire ("Le plus grand élément est ", T(iMax, jMax))

Ecrire ("Il se trouve aux indices ", iMax, "; ", jMax)

Achtaich Khadija



Tableaux à n dimensions

- Si je déclare un tableau $T(3, 5, 4, 4)$, il s'agit d'un espace mémoire contenant $3 \times 5 \times 4 \times 4 = 240$ valeurs. Chaque valeur y est repérée par quatre coordonnées.
- L'utilisation de tableaux à plus de 3 dimensions reste rare sauf dans des programmes de mathématiques ou à base de mathématiques.

Achtaich Khadija



Tableaux à n dimensions

Exemple:

supposons que nous voulons écrire un programme permettant de saisir les notes de tous les étudiants d'une école.

- Cet école dispose de 4 classes
- Chaque classe comporte 20 étudiants.
- Chaque étudiant à 3 notes.

Achtaich Khadija



Tableaux à n dimensions

Exemple:

Tableau $T(4,20,3)$:réel

Pour $i \leftarrow 0$ à 3

Pour $j \leftarrow 0$ à 19

Pour $k \leftarrow 0$ à 2

Ecrire(" donner la note ", k , " de l'étudiant numéro ", j , " de la classe ", i)

Lire($T(i,j,k)$)

FinPour k

FinPour j

FinPour

Achtaich Khadija



Recherche dans un tableau

Exemple introductif:

Écrire un algorithme permettant de chercher un nom donné dans un tableau de noms.

Achtaich Khadija



Recherche dans un tableau

Exemple introductif:

Une erreur courante chez les programmeurs débutants est d'écrire un algorithme comme le suivant :

Achtaich Khadija



Recherche dans un tableau

Algorithme recherche

Tableau T(20) : Chaîne de caractère

Variable nom : Chaîne de caractère

Début

//nous supposons que la tableau est déjà rempli

Ecrire("donner le nom à rechercher")

Lire (nom)

Pour $i \leftarrow 0$ à 19

Si (nom = T(i))

Ecrire("nom fait partie du tableau")

Sinon

Ecrire(" nom ne fait pas partie du tableau")

Finsi

Finpour i

Fin

Achtaich Khadija



Recherche dans un tableau

- L'exécution de cet algorithme donne des message comme les suivants:

Nom ne fait pas partie du tableau

Nom fait partie du tableau

...

Nom ne fait pas partie du tableau

Nom ne fait pas partie du tableau

Nom fait partie du tableau

Achtaich Khadija



Recherche dans un tableau

- Or nous avons besoin d'avoir le message une seule fois a la fin pour nous dire si la valeur recherchée dans le tableau y existe ou pas.
- Donc le message doit être écrit à l'extérieur de la boucle lorsque le balayage du tableau est entièrement accompli.
- Nous avons donc besoin d'une variable booléenne pour nous informer de la présence ou non de l'élément recherché.
- Cette variable est appelée **flag** ou drapeau.

Achtaich Khadija



Recherche dans un tableau

notion de Flag

- La recherche dans un tableau est une activité récurrente en programmation.
- Elle possède aussi des techniques bien connues qu'il faudra connaître et maîtriser comme la technique du flag ou drapeau.

Achtaich Khadija



Recherche dans un tableau

- L'algorithme ressemblera donc a :

Début

Ecrire "Entrez le nom à rechercher"

Lire (Nom)

Pour $i \leftarrow 0$ à 19

????????????????????

????????????????????

Finpour i

Si (Trouve)Alors

Ecrire ("Nom fait partie du tableau")

Sinon

Ecrire ("Nom ne fait pas partie du tableau")

FinSi

Fin

Achtaich Khadija



Recherche dans un tableau

Il ne nous reste plus qu'à gérer la variable

Trouve.

- Ceci se fait en deux étapes:
 - un test figurant dans la boucle, indiquant lorsque la variable Trouve doit devenir vraie (à savoir, lorsque la valeur Nom est rencontrée dans le tableau). Et attention : le test est asymétrique. Il ne comporte pas de "sinon".
 - La valeur de départ de la variable Trouve doit être Faux.

Achtaich Khadija



Recherche dans un tableau

Ainsi le corps de algorithme devient :

Début

Ecrire ("Entrez le nom a rechercher")

Lire (Nom)

Trouve \leftarrow Faux

Pour $i \leftarrow 0$ à 19

Si (Nom = Tab(i)) Alors

Trouve \leftarrow Vrai

FinSi

Finpour i

Si Trouve Alors //Équivalent à : Si
Trouve=Vrai

Ecrire (" Nom fait partie du tableau")

Sinon

Ecrire (" Nom ne fait pas partie du tableau")

FinSi

Fin

Achtaich Khadija



Recherche dans un tableau

Or on remarque que l'algorithme parcourt tout le tableau même si le nom se trouve dans la première case du tableau.

L'algorithme suivant présente une solution à ce problème

Achtaich Khadija



Recherche dans un tableau

```
Début
Ecrire ("Entrez le nom a rechercher")
Lire (Nom)
Trouve ← Faux
i ← 0
Tantque((non trouve) et( i<19))
Si (Nom = Tab(i)) Alors
    Trouve ← Vrai
FinSi
i←i+1
FinTantque

Si ( Trouve) Alors //Équivalent à : Si Trouve=Vrai
    Ecrire (Nom ," fait partie du tableau")
Sinon
    Ecrire (Nom, " ne fait pas partie du tableau")
FinSi
Fin
```

Achtaich Khadija



Tri par Sélection

Le tri par sélection

- Le tri par sélection est le plus intuitive des algorithmes de tri.
- Son principe est comme suit:
 - On recherche l'élément le plus petit d'un tableau.
 - On le permute avec l'élément qui se trouve à la première position du tableau.
 - On répète cette opération en commençant à partir de la deuxième case du tableau, etc., jusqu'à ce que tout le tableau soit trié.

Achtaich Khadija



Tri par Sélection

Exemple :

- Supposons que nous voulons trier le tableau suivant dans un ordre croissant.

1	2	3	4	5	6	7
7	15	2	17	40	5	8

- Le plus petit élément du tableau est le nombre « 2 » qui se trouve à la case numéro 3.
- Nous l'échangeons donc avec la case numéro 1 du tableau.

1	2	3	4	5	6	7
2	15	7	17	40	5	8

Achtaich Khadija

Tri par Sélection

- Maintenant que le plus petit élément du tableau est à la première case nous nous concentrons sur la tableau à partir de la deuxième case (c'est comme si la première case n'existe pas).

Partie triée		Partie non triée				
1	2	3	4	5	6	7
2	15	7	17	40	5	8

- Dans le reste du tableau à partir de la deuxième case, le plus petit élément est « 5 » qui se trouve à la case numéro 6.
- On procède donc à son échange avec la deuxième case du tableau.

Partie triée		Partie non triée				
1	2	3	4	5	6	7
2	5	7	17	40	15	8

Achtaich Khadija



Tri par sélection

Algorithme triParselection

Variable i, j, posMin : entier

Tableau $T()$: réel

Variable temp : réel

Début

//nous supposons que la tableau est déjà rempli

Pour $i \leftarrow 0$ à $n-2$

$\text{posMin} \leftarrow i$

 Pour $j \leftarrow i+1$ à $n-1$

 si $(T(\text{posMin}) > T(j))$

$\text{posMin} \leftarrow j$

 FinSi

 Finpour j

$\text{temp} \leftarrow T(i)$

$T(i) \leftarrow T(\text{posMin})$

$T(\text{posMin}) \leftarrow \text{temp}$

 Finpour i

Fin

Achtaich Khadija



Le tri par bulles

Tri de tableau + flag = tri à bulles

Idée de base :

- On prend chaque élément du tableau et on le compare avec qui le suit.
- Si l'ordre n'est pas bon, on permute ces deux éléments.
- Et on recommence jusqu'à ce que l'on n'ait plus aucune permutation à effectuer.

Achtaich Khadija



Le tri par bulles

Dans un tri croissant (ou décroissant) les éléments les plus grands (ou petits) remontent ainsi peu à peu vers les dernières places, ce qui explique la dénomination de « tri à bulle ».

On ne sait jamais par avance combien de remontées de bulles (balayage) on doit effectuer. Ceci implique l'utilisation d'un flag qui nous dira si dans le dernier balayage du tableau nous avons effectué une permutation.

Si c'est le cas on recommence sinon le tableau est trié.

Achtaich Khadija



Le tri par bulles

Nous utilisons le flag permute, La boucle principale sera alors :

Variable permute:Booléen

Début

...

TantQue (permute)

.....

FinTantQue

Fin

Achtaich Khadija



Le tri par bulles

Question : Que va-t-on faire à l'intérieur de la boucle ?

Réponse : Prendre les éléments du tableau, du premier jusqu'à l'avant dernier, et procéder à un échange si nécessaire.

➤ Mais il ne faut pas oublier un détail capital : la gestion de notre flag. L'idée, c'est que cette variable va nous signaler le fait qu'il y a eu au moins une permutation effectuée.

➤ Il faut donc :

- lui attribuer la valeur Vrai dès qu'une seule permutation a été faite (il suffit qu'il y en ait eu une seule pour qu'on doive tout recommencer encore une fois).
- la remettre à Faux à chaque tour de la boucle principale (quand on recommence un nouveau tour général de bulles, il n'y a pas encore eu d'éléments échangés),
- dernier point, il ne faut pas oublier de lancer la boucle principale, et pour cela de donner la valeur Vrai au flag au tout début de l'algorithme

Achtaich Khadija



Le tri par bulles

La solution complète donne donc :

Variable permute: Booléen

Début

.....

permute ← Vrai

TantQue (permute)

permute ← Faux

Pour $i \leftarrow 0$ à 8

Pour $i \leftarrow 0$ à 8

Si ($t(i) > t(i+1)$) Alors

temp ← $t(i)$

$t(i) \leftarrow t(i+1)$

$t(i+1) \leftarrow$ temp

permute ← Vrai

Finsi

FinPour

FinTanque

Fin

Achtaich Khadija



Le tri par bulles

Exemple

1	3	2	5
---	---	---	---

Achtaich Khadija



Les tableaux dynamiques

Contexte

La déclaration d'un tableau se fait avec la connaissance préalable de nombre de ses éléments.

Que faire si ce nombre ne sera connu qu'au moment de l'exécution ?

Déclarer un tableau avec un très grand nombre d'éléments (100000....) alors que dans l'exécution seuls quelques uns sont utilisés serait un gaspillage inutile de la mémoire.

Achtaich Khadija



Les tableaux dynamiques

Solution:

L'algorithmique nous permet de déclarer un tableau sans préciser le nombre de ses éléments.

Ce n'est qu'au moment de l'exécution qu'on le fait après avoir pris connaissance de ce nombre.

Dans ce cas ce nombre est précisé avec l'instruction Redim qui permet ainsi de redimensionner un tableau.

Achtaich Khadija



Les tableaux dynamiques

Exemple:

Algorithme tabDynamique

Tableau Notes() :entier

Variable nb :entier

Debut

Ecrire ("Combien y a-t-il de notes à saisir ? ")

Lire nb

Redim Notes(nb)

...

Achtaich Khadija



Les tableaux dynamiques

Application:

- Écrivez un algorithme calculant la somme des valeurs d'un tableau

Algorithme sommeTabDynamique

Tableau T() :entier

Variable nb, i, Som: entier

Debut

Ecrire ('Combien y a-t-il de valeurs dans le tableau ?')

Lire (nb)

Redim T(nb)

/*saisie des valeurs du tableau*/

Pour i \leftarrow 0 à nb-1

Ecrire ("Entrer la valeur de T(",i, ") ")

Lire(T(i))

Achtaich Khadija



Les tableaux dynamiques

/*calcul de la somme des éléments du tableau*/

Som \leftarrow 0

Pour $i \leftarrow 0$ à $nb-1$

Som \leftarrow Som + T(i)

FinPour

/*affichage de la somme*/

Ecrire (« Somme des éléments du tableau : « , Som)

Fin

Achtaich Khadija



Les tableaux dynamiques

Exercices d'application

Réécrire l'algorithme de calcul de la moyenne lorsque le nombre de valeurs tapées par l'utilisateur n'est pas connu à l'avance. Ce nombre sera saisi par l'utilisateur lui-même avant de saisir l'ensemble des valeurs.

Achtaich Khadija



Les tableaux dynamiques

Algorithme moyenne

Tableau Valeurs() : **Entier**

Variables i, NbElements, Somme : **Entier**

Variable Moyenne : **Réel**

Début

Ecrire ("Veuillez saisir le nombre total des éléments")

Lire(NbElements)

Redim Valeurs(NbElements)

Pour i \leftarrow 0 **à** NbElements – 1 **Faire**

Ecrire ("Veuillez saisir la note numéro " , i)

Lire(Valeurs(i))

Fin Pour

Achtaich Khadija



Les tableaux dynamiques

Somme $\leftarrow 0$

Pour $i \leftarrow 0$ **à** NbElements - 1 **Faire**

 Somme \leftarrow Somme + Valeurs(i)

Fin Pour

Moyenne \leftarrow Somme / NbElements

Ecrire ("La moyenne est " , Moyenne)

Fin

Achtaich Khadija

