

Cours d'algorithmique Structures et Tableaux à plusieurs dimensions - EISTI - ING 1

Ecole Internationale des Sciences du Traitement de l'Information

Définition

- ▶ Type complexe défini par l'utilisateur
- ▶ Composé d'un ensemble de champs de différents types

Syntaxe

```
type nomType = structure
  nomChamps1 : T1
  nomChamps2 : T2
  ...
  nomChampsN : TN
finstructure
```

T_1, T_2, \dots, T_N sont des types de bases ou eux mêmes définis

Exemples de structures

Structure de types simples

```
type personne = structure
  nom : chaine
  prenom : chaine
  age : entier
finstructure
```

Structure de types complexes

```
type etudiant = structure
  identite : personne
  numero : entier
  annee : entier
  matieres : tableau(N): chaines
  nbmatieres : entier
finstructure
```

Déclaration et accès

Déclaration

```
type nomType = structure
  ...
  nomChampsi
  ...
finstructure
variables :
  nomVariable : nomType
```

Accès

```
nomVariable.nomChampsi
```

Afficher les matières d'un étudiant donné

```
procedure afficherMatiere(E e : etudiant)
  variables
    i : entier
  i ← 1
  tant que i ≤ e.nbmatieres faire
    ecrireLn(e.matiere(i))
    i ← i+1
  fin tant que
fin procedure
```

Tableaux plusieurs dimensions

- ▶ Simple vue de l'esprit. En mémoire interne les données sont toujours représentés de manière contigüe.
- ▶ Simple à représenter pour dimension 2 : repère géométrique, matrice, échiquier, labyrinthe ...
- ▶ Beaucoup plus délicat pour les dimensions supérieures : espace (3), hyperspace(n).

Mise en oeuvre

- ▶ Mélange de tableaux à une dimension et de structures :
 - ▶ Un tableau de tableau de tableau de ... de tableau d'éléments
 - ▶ Parcours de cette structure via des boucles imbriquées : degré d'imbrication = dimension du tableau
 - ▶ Coût du parcours obtenu directement en fonction de ce degré d'imbrication (dimension $k = n^k$ itérations pour parcourir une structure de taille n).
 - ▶ En général, algorithmes praticables jusqu'à l'ordre 2 (algorithme des graphes ALGO2, calcul matriciel ANANUM) voire 3 (compilation THLANG), rarement au delà.

Tableau a 2 dimensions d'entiers :
Déclaration :

```
variables :  
t: tableau(N,M): entiers
```

Accès et affectation :

```
t(i,j) ← 0
```


Parcours

```
pour i ← 1 a n pas 1  
  pour j ← 1 a m pas 1  
    traiter(t(i,j))  
  fpour  
fpour
```

Gestion circulaire : tores

- ▶ Algorithmique modulaire
 - ▶ Case suivant la dernière case de la ligne = première case de la ligne (et vice-versa)
 - ▶ Case suivant la dernière case de la colonne = première case de la colonne (et vice-versa)
- ▶ Faire attention a la gestion des indices

Exemples

- ▶ Calcul matriciel : systèmes linéaires
- ▶ Jeux de plateaux : dames, échec, ...
- ▶ Jeux de labyrinthes : Pac-Man (circulaire), solitaire, ...
- ▶ Affichages en tous genres : images = tableaux de pixels
- ▶ Algorithmes des graphes : réseaux, IA, ...