



Traitement séquentiel de tableaux à deux dimensions

Jean-Michel Adam - UGA - UFR SHS

Tableaux à deux dimensions

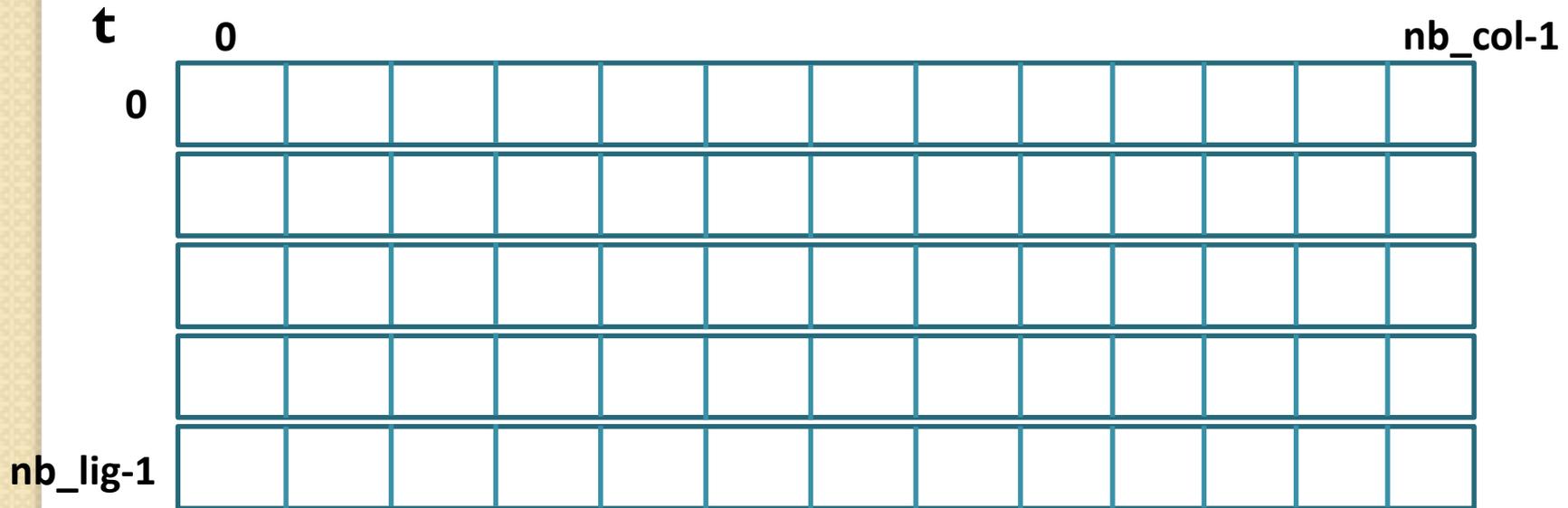
- **t** : tableau sur $[bi_v..bs_v, bi_h..bs_h]$ de typedeselements
- Analogie : une grille, une matrice

t

	bi_h												bs_h
bi_v													
bs_v													

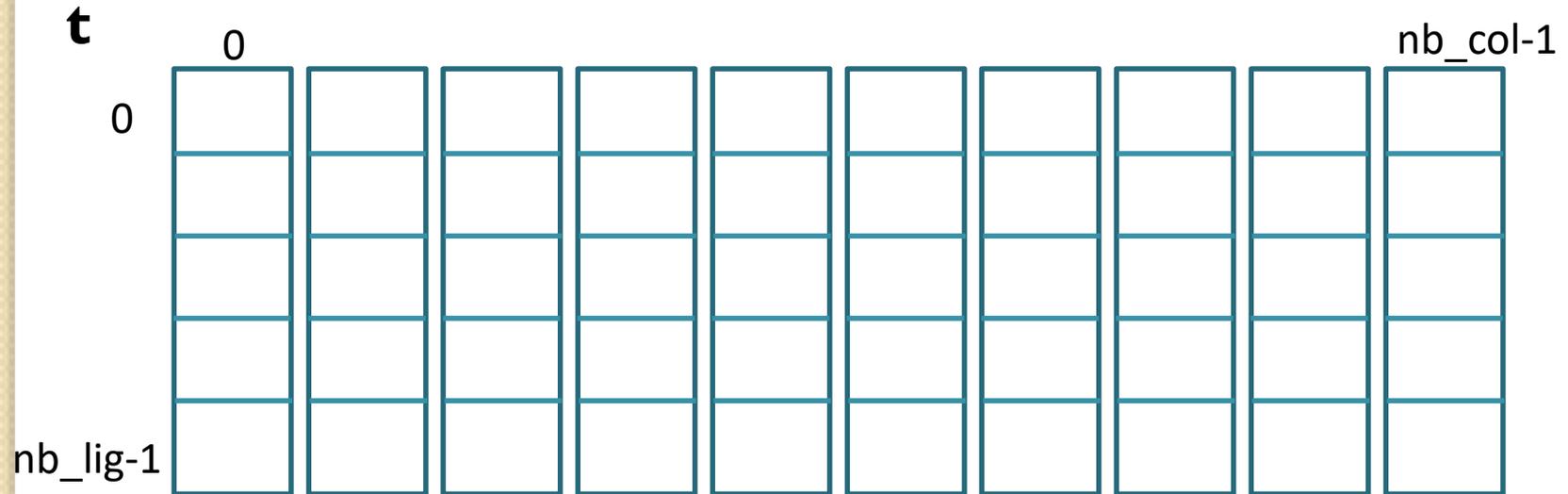
Tableaux à deux dimensions

- **t** : tableau sur $[0..nb_lig-1]$ de tableaux sur $[0..nb_col-1]$ de typedeselements
- Analogie : tableau de lignes



Tableaux à deux dimensions

- **t** : tableau sur $[0..nb_lig-1]$ de tableaux sur $[0..nb_col-1]$ de typedeselements
- Analogie : tableau de colonnes



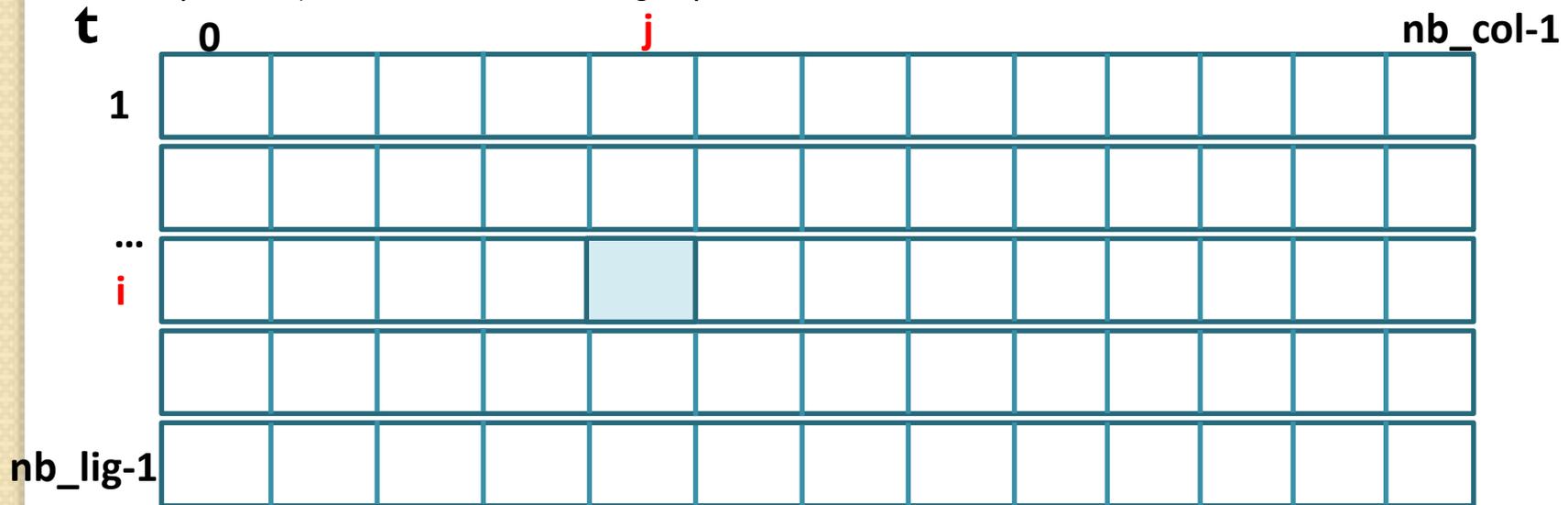
Tableaux à deux dimensions

- Désignation d'un élément :

$t[i][j]$ désigne

- l'élément d'indice j de dans le tableau $t[i]$ (la ligne i)
- l'élément d'indice i dans la colonne j

On place toujours d'abord l'indice de ligne, puis l'indice de colonne



Tableaux à deux dimensions

- Désignation d'un élément :
 $T[i, j]$ désigne
 - l'élément de coordonnées $[i, j]$ dans le grille t
c'est-à-dire l'élément de ligne i et de colonne jOn place toujours d'abord l'indice de ligne, puis l'indice de colonne

The diagram shows a 2D array grid with 5 rows and 12 columns. The grid is labeled with t at the top left. The top row is labeled with 0 at the left and nb_col-1 at the right. The top-left cell is labeled with 0 at the left and 0 at the top. A red i is placed to the left of the third row, and a red j is placed above the fifth column. The cell at the intersection of row i and column j is shaded light blue.

t	0			j								nb_col-1
0												
nb_lig-1												

Tableaux à deux dimensions

- Enumération des éléments
 - En général, le **parcours** d'un tableau se fait à l'aide de deux itérations emboîtées.
 - On parcourt une suite de lignes ou une suite de colonnes.
 - Le traitement de la ligne (colonne) courante consiste à énumérer et traiter les éléments de cette ligne (colonne).

Parcours d'une séquence de lignes

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // **indice de parcours des lignes de t**

j : entier entre 0 et NB_COL // **indice de parcours des éléments de t[i]**

// on suppose que le tableau est plein

Algorithme

i ← 0 Démarrer

Initialisation du traitement de t

tantque **i < NB_LIG** faire → Fin de séquence

j ← 0

initialisation du traitement de t[i]

tantque j < NB_COL faire

 traitement de t[i][j]

 j ← j + 1

ftantque

 terminaison du traitement de t[i]

i ← i + 1 Avancer

ftantque

Terminaison du traitement de t

Traitement de l'élément courant
ligne i notée t[i]

Parcours d'une séquence de lignes

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des éléments de t[i]

// on suppose que le tableau comporte l lignes et c colonnes remplies

Algorithme

$i \leftarrow 0$ Démarrer

Initialisation du traitement de t

tantque $i < l$ faire \rightarrow Fin de séquence

$j \leftarrow 0$

initialisation du traitement de t[i]

tantque $j < c$ faire

 traitement de t[i][j]

$j \leftarrow j + 1$

ftantque

 terminaison du traitement de t[i]

$i \leftarrow i + 1$ Avancer

ftantque

Terminaison du traitement de t

Traitement de l'élément courant
ligne i notée t[i]

Parcours d'une séquence de colonnes

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type elements

i : entier entre 0 et NB_LIG // indice de parcours des éléments de t[][j]

j : entier entre 0 et NB_COL // indice de parcours des colonnes de

// on suppose que le tableau est plein

Algorithme

j ← 0

Démarrer

Initialisation du traitement de t

tantque j < NB_COL faire → Fin de séquence

i ← 0

initialisation du traitement de t[][j]

tantque i < NB_LIG faire

 traitement de t[i][j]

 i ← i + 1

ftantque

terminaison du traitement de t[][j]

j ← j + 1

Avancer

ftantque

Terminaison du traitement de t

Traitement élément courant
(colonne j notée t[][j])

Parcours d'une séquence de colonnes

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type elements

i : entier entre 0 et NB_LIG // indice de parcours des éléments de t[i][j]

j : entier entre 0 et NB_COL // indice de parcours des colonnes de

// on suppose que le tableau comporte l lignes et c colonnes remplies

Algorithme

j ← 0

Démarrer

Initialisation du traitement de t

tantque j < c faire

→ Fin de séquence

i ← 0

initialisation du traitement de t[i][j]

tantque i < l faire

 traitement de t[i][j]

 i ← i + 1

ftantque

terminaison du traitement de t[i][j]

j ← j + 1

Avancer

ftantque

Terminaison du traitement de t

Traitement élément courant
(colonne j notée t[i][j])

Tableaux à deux dimensions

- Enumération des éléments
 - Il est également possible de parcourir le tableau à **l'aide d'une seule itération** plutôt que de deux itérations emboîtées
 - Intéressant pour la recherche d'un élément dans le tableau
 - On considère le tableau comme une suite d'éléments, l'élément qui suit le dernier de la ligne k est le premier élément de la ligne $k+1$ (parcours ligne/ligne)

Enumération des éléments de la table t ligne par ligne (tableau plein)

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des colonnes de t

EC	t[i][j]
FDS	i = NB_LIG
AV	<u>selon</u> j j < NB_COL-1 : j ← j + 1 j = NB_COL-1 : j ← 0 ; i ← i + 1 <u>fselon</u>
DEM	i ← 0 ; j ← 0

Enumération des éléments de la table t colonne par colonne (tableau plein)

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des colonnes de t

EC	t[i][j]
FDS	j = NB_COL
AV	<u>selon</u> i i < NB_LIG-1 : i ← i + 1 i = NB_LIG-1 : i ← 0 ; j ← j + 1 <u>fselon</u>
DEM	j ← 0 ; i ← 0

Parcours des éléments de t (ligne / ligne)

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des colonnes de t

Algorithme

$i \leftarrow 0 ; j \leftarrow 0$ Démarrer

Initialisation du traitement de t

tantque $i < \text{NB_LIG}$ faire \rightarrow Fin de séquence

traitement de $t[i][j]$ Traitement de l'élément courant $t[i][j]$

selon j

j < NB_COL-1 : $j \leftarrow j + 1$

j = NB_COL-1 : $j \leftarrow 0 ; i \leftarrow i + 1$

fselon

Avancer

ftantque

Terminaison du traitement de t

Recherche d'un élément vérifiant P (ligne / ligne)

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des colonnes de t

Algorithme

i ← 0 ; j ← 0 Démarrer

tantque i < NB_LIG et puis ¬ P(t[i][j]) faire → Fin de séquence

selon j

 j < NB_COL-1 : j ← j + 1

 j = NB_COL-1 : j ← 0 ; i ← i + 1

fselon

Avancer

ftantque

selon i

 i < NB_LIG : // t[i][j] vérifie P

 i = NB_LIG : // aucun élément de t ne vérifie P

fselon

Recherche ligne/ligne (itérations emboîtées)

Lexique

t : tableau sur [0..NB_LIG-1] de tableaux sur [0..NB_COL-1] de type_elements

i : entier entre 0 et NB_LIG // indice de parcours des lignes de t

j : entier entre 0 et NB_COL // indice de parcours des éléments de t[i]

trouvé : booléen // inter : vrai si on a trouvé un élément vérifiant P

// on suppose que le tableau comporte l lignes et c colonnes remplies

Algorithme

i ← 0 ; trouvé ← faux

tantque i < l et non trouvé faire

 j ← 0

tantque j < c et puis non P(t[i][j]) faire

 j ← j + 1

ftantque

 // j = c ou alors P(t[i][j])

 trouvé ← j < c

 i ← i + 1

ftantque

// i = l ou trouvé

si trouvé

alors // t[i-1][j] vérifie P

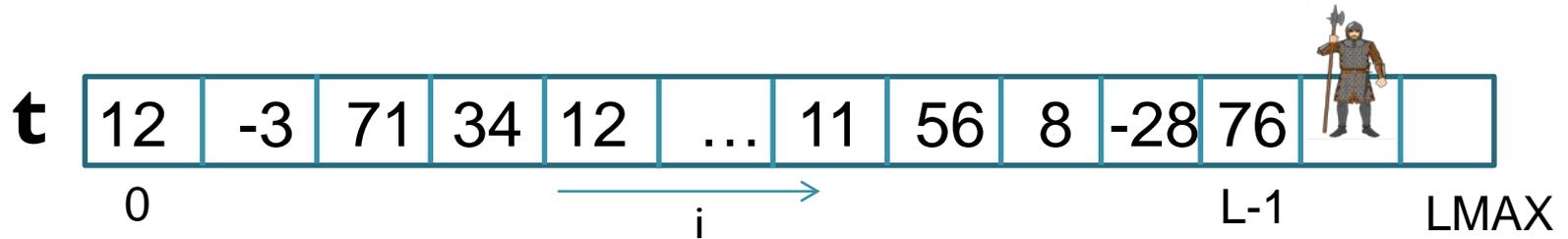
sinon // aucun élément de t ne vérifie P

fsi

Technique de la sentinelle



- Recherche dans un tableau



- Idée : placer une valeur recherchée à la fin de la séquence
 - Ainsi on est sûr que la recherche réussit
 - On peut appliquer le schéma de recherche simplifié

Technique de la sentinelle



Lexique

t : tableau sur [0..LMAX] d'entiers

L : réel ≥ 0 // longueur de la séquence

i : entier sur 0..LMAX // indice de recherche

Algorithme

i \leftarrow 0

T[L] \leftarrow v // v est une valeur vérifiant P (valeur sentinelle)

tantque non P(T[i]) faire

 i \leftarrow i + 1

ftantque

// P(T[i]) est vrai, si $i < L$ alors on a trouvé un élément vérifiant P

selon i

i < L : // trouvé

i = L : // non trouvé

fselon

