

SAE - Guide : grilles stockées en tableaux 1D ou 2D ?

Fonctionnement

Pour une grille de $n=3$ colonnes et $m=4$ lignes

2D: deux coordonnées x,y

0,0	1,0	2,0
0,1	1,1	2,1
0,2	1,2	2,2
0,3	1,3	2,3

1D: une coordonnée z

0	1	2
3	4	5
6	7	8
9	10	11

Conversion

2D -> 1D: * $z=x+n*y$

1D -> 2D: * $x=z \% n$ * $y=z / n$

Allocation mémoire

2D: $m+1$ tableaux: m (un par ligne) de taille n , et 1 de taille m pour stocker tous les tableaux de lignes (donc $m+1$ `malloc`). Mémoire non contigüe \Rightarrow perte de performances.

1D: 1 tableau de taille $m*n$

Parcours

2D: Deux boucles for imbriquées

1D: Une seule boucle fort

Indices

2D: Stockés dans deux variables, ou une structure $\{x,y\}$ (structure nécessaire pour des returns ou stocker des tableaux d'indices)

1D: Un seul entier

Entrées/Sorties

Elles correspondent au parcours 1D: dans les deux cas c'est directement un parcours du tableau

Vérification des bords

2D: Tester $x==0$, $x==n-1$, $y==0$, $y==m-1$

1D: Conversion de z vers (x,y) nécessaire

Coordonnées des voisins

2D Clair: $(x-1,y)$; $(x,y+1)$; $(x-1,y-1)$

1D Plus complexe: $z-1$, $z+n$, $z-n-1$

Version hybride

- On stocke un tableau 1D
- On utilise deux coordonnées
- On réalise une conversion à chaque accès (avec des fonctions et **get** et **set**)

Bilan

2D pure: déconseillée pour les problèmes d'allocation mémoire

1D pure: le plus simple pour la gestion de la mémoire et les entrées sorties, mais faire attention aux opérations sur les bords et les voisins

Hybride: Bien préparer les fonctions annexes (get et set, structure pour stocker le tableau), mais simplifie les déplacements dans la grille.