

## Tableaux, itérations et invariants

Christophe Fiorio

### \* Exercice 1 Tableaux convexes

On considère des tableaux à une dimension, à  $n$  éléments booléens. Un tel tableau est *convexe* si tous les éléments du tableau ayant la valeur `true` sont consécutifs. Par exemple, `[false, true, true, true, false, false, false, false]`, `[false, false, false, false, false, true, true, true]` sont convexes tandis que `[true, true, false, true, false, false, false, false]` ne l'est pas.

- proposez un invariant<sup>1</sup> permettant de vérifier la propriété de convexité;
- à partir de l'invariant, écrire une fonction `Est_Convexe`, qui, étant donné un tableau retourne vrai si ce tableau est convexe, faux sinon.

### \* Exercice 2 Recherche du sous-tableau maximum

On considère un tableau  $M$  à  $n$  éléments de type `Entier`. Une "tranche" de tableau  $(i, j)$  correspond aux éléments du tableau dont l'indice  $k$  vérifie :  $i \leq k \leq j$ . La "somme d'une tranche"  $(i, j)$  est la somme des éléments de la tranche.

- Écrire un invariant qui exprime le résultat recherché.
- En déduire l'algorithme d'une fonction `Somme_Max` qui pour un tableau  $M$  donné, calcule le maximum, sur toutes les tranches possibles, de la somme des tranches.

*Remarque* : il y a plusieurs algorithmes possibles, d'efficacité variable; évaluez l'efficacité de votre algorithme (en nombre de boucle imbriquées). NB : il y a un algorithme qui donne la solution en 1 simple boucle.

### \*\*\* Exercice 3 Le problème du drapeau hollandais

Dans cet exercice on considère un tableau de caractères  $T$ , de taille  $n$ , avec une restriction : les valeurs possibles des éléments du tableau sont limitées à l'ensemble `{'B', 'W', 'R'}`.

Rappel : on a `'B' < 'W' < 'R'`.

Écrire une procédure qui réordonne les éléments du tableau  $T$  de telle manière qu'après, les éléments de  $T$  soient dans l'ordre des caractères.

▲ **attention** : votre procédure doit effectuer un seul parcours du tableau; vous ne pouvez donc pas utiliser les algorithmes de tri classiques.

### \*\*\* Exercice 4 Matrices de booléens

Dans cet exercice, on considère des tableaux de booléens à deux dimensions, de taille  $n \times m$ . On appellera *rectangle* d'un tableau  $M$  de type `matrice` une zone du tableau définie par un quadruplet  $(lig1, col1, lig2, col2)$  tel que :  $lig1 \leq lig2$ ,  $col1 \leq col2$ ,  $0 \leq lig1, lig2 \leq n$  et  $0 \leq col1, col2 \leq m$  : il s'agit du rectangle dont le coin supérieur gauche est la case  $(lig1, col1)$  et le coin inférieur droit est la case  $(lig2, col2)$ . Par exemple, sur la figure,  $(2, 1, 5, 3)$  définit un rectangle.

Un rectangle est "*vide*" quand toutes ses cases ont la valeur `false`. Dans la figure, les cases ayant la valeur `true` sont en grisé; le rectangle défini par  $(2, 1, 5, 2)$  est *vide*, les rectangles définis par  $(2, 1, 5, 5)$  et  $(2, 4, 4, 5)$  ne sont pas vides. La *surface* d'un rectangle est le nombre de cases qu'il contient. Par exemple, la surface du rectangle  $(2, 1, 5, 2)$  est 8.

1. Pour rappel, un invariant n'est rien d'autre qu'une ou plusieurs propriétés exprimant le résultat en cours d'exécution de l'algorithme, c'est à dire le résultat à une itération quelconque, qui permet de prouver le résultat final une fois l'itération terminée.

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								

- Écrire un sous-programme qui étant donné un tableau  $M$  et un rectangle indique si le rectangle est vide .
- Écrire un sous-programme qui étant donné un tableau  $M$  et une case d'indices  $(i, j)$  ( $i$  est l'indice de la ligne,  $j$  est l'indice de la colonne) de ce tableau retourne la surface d'un **rectangle vide de surface maximum** de  $M$  dont la case  $(i, j)$  est le coin supérieur gauche. Réfléchissez à plusieurs solutions. Vous pouvez bien sûr être amenés à définir d'autres sous-programmes.

Sur la figure, pour la case  $(2,1)$  ce sous-programme retourne la valeur 12 qui correspond aux deux rectangles  $(2,1,4,4)$  et  $(2,1,5,3)$ .