

## Manipulation de tableaux

Christophe Fiorio

On souhaite implémenter un type concret `TInt` dont le type abstrait est le suivant :

- 
- `init : Int × Int → TInt` // `init(a,n)` crée un tableau de `n` éléments initialisés avec la valeur `a`
  - `count : TInt → Int` // donne le nombre d'éléments du tableau ;  
// `count(init(a,n))==n`
  - `[] : TInt × Int → Int` // retourne l'élément à l'indice donné en paramètre
  - `[] : TInt × Int × Int → TInt` // change la valeur de l'entier à l'indice donné par l'entier donné en paramètre  
// `T=init(a,n) ⇒ ∀ 0 ≤ j < n, T[j]==a ; T[j]=k ⇒ T[j]==k`
  - `contains : TInt × Int → bool` // True si l'entier donnée en paramètre appartient au tableau ; `contains(T,a) ⇒ ∃ i, T[i]==a`
  - `firstIndex : TInt × Int → Int|Vide` // indice de la première valeur (dans l'ordre des indices) donnée en paramètre  
// `firstIndex(T,a)==i ⇒ T[i]==a et ∀ 0 ≤ j < i, T[j]≠a`  
// `firstIndex(T,a)==Vide ⇒ ∀ 0 ≤ i < n, T[i]≠a`
  - `lastIndex : TInt × Int → Int|Vide` // indice de la dernière valeur (dans l'ordre des indices) donnée en paramètre  
// `lastIndex(T,a)==i ⇒ T[i]==a et ∀ i < j < n, T[j]≠a`  
// `lastIndex(T,a)==Vide ⇒ ∀ 0 ≤ i < n, T[i]≠a`
  - `nbOccur : TInt × Int → Int` // nombre d'occurrences de la valeur passée en paramètre  
// `nbOccur(T,a)==p ⇒ ∃ i0, ..., ip tel que ∀ ik, 0 ≤ k ≤ p, ik ∈ [0, n-1] et T[ik]==a`
  - `min : TInt → Int` // plus petite valeur de T, `min(T)==m ⇒ ∀ 0 ≤ i < n, T[i] ≥ m`
  - `max : TInt → Int` // plus grande valeur de T, `max(T)==m ⇒ ∀ 0 ≤ i < n, T[i] ≤ m`
- 

### Exercice 1 structure de données

Proposez une structure de données pour le type abstrait `TInt`

### Exercice 2 algorithmes des fonctions

Implémentez les fonctions du type, en prouvant à chaque fois vos algorithmes et en évaluant leur complexité.

### \* Exercice 3 Ordre lexicographique

Écrire un algorithme `Compare` qui, à partir de deux tableaux `T1` et `T2` à `n` éléments, a pour résultat :

- la valeur -1 si tous les éléments de `T1` sont respectivement avant les éléments de `T2` dans l'ordre lexicographique.  
*Exemple : T1=[1,2,1] et T2=[1,2,3]*
- la valeur 0 si `T1` et `T2` sont égaux  
*Exemple : T1=[4,2,5] et T2=[4,2,5]*
- la valeur 1 sinon.  
*Exemple : T1=[1,4,1] et T2=[1,2,3]*

### \*\* Exercice 4 Recherche de sous-tableau

Écrire un algorithme `isSub` qui, étant donné deux tableaux `T1` et `T2`, indique si `T1` est sous-tableau de `T2`

- `isSub : TInt × TInt → bool` // `isSub(T1,T2)` retourne true si `T1` est sous-tableau de `T2`, et false sinon. `T1` est sous-tableau de `T2` si

```
//  $\exists 0 \leq k < \text{count}(T2)$ , tel que  $\forall 0 \leq i < \text{count}(T1), T1[i] == T2[k + i]$   
// et tel que  $k + \text{count}(T1) \leq \text{count}(T2)$ 
```

Exemples :

- `isSub([3,4,5],[1,2,3,4,5,6]) == true`
- `isSub([3,4,5],[1,3,2,4,5,6]) == false`
- `isSub([3,4,5],[1,3,4]) == false`