

**TD3 : programmation dynamique**

On s'intéresse à l'écriture d'une fonction de comparaison de listes de matricules (tous écrits sur 3 chiffres) ordonnées de manière croissante. On définit la relation d'ordre partielle  $\leq$  qui indique si une liste  $x$  de matricules est inférieure à une  $y$ . Pour deux listes  $x = x_1, \dots, x_n$  et  $y = (y_1, \dots, y_m)$  on dit que  $x < y$  si :

- soit  $x$  est vide
- soit  $x$  est non vide et il existe  $i \in 1..n$  et  $j \in 1..m$  tels que :
  - $x_i \leq y_j$
  - $(x_1, \dots, x_{i-1}) \leq (y_1, \dots, y_{j-1})$
  - $(x_{i+1} \dots x_n) \leq (y_{j+1}, \dots, y_m)$

Par exemple, si on prend  $x = (102, 123, 224, 334, 335, 916)$  et  $y = (102, 245, 253, 467)$  alors on a  $y \leq x$  mais pas  $x \leq y$ .

On utilisera dans la suite les définitions suivantes :

```
type CHIFFRE = 0..9;
type MATRICULE = array [1..3] of CHIFFRE;
type LISTE_MATRICULES = array of MATRICULE;
function inferieurOuEgal (a, b : MATRICULE) : BOOLEAN;
```

**Q 1** Ecrire une fonction récursive `function inferieur (x, y : LISTE_MATRICULES) : BOOLEAN;`

**Q 2** Identifiez un meilleur des cas pour la comparaison. Quelle est alors la complexité en nombre de comparaison de matricules ?

**Q 3** Quelle est la complexité asymptotique dans le pire des cas ?

Le problème que nous traitons se prête bien à une solution par programmation dynamique. En effet, on peut réécrire la définition de  $\leq$  comme :  $x = (x_1, \dots, x_n) \leq y = (y_1, \dots, y_m)$  si :

- soit  $x$  est vide ( $n = 0$ )
- soit  $(x_1, \dots, x_{n-1}) \leq (y_1, \dots, y_{m-1})$  et  $x_n \leq y_m$
- soit  $(x_1, \dots, x_n) \leq (y_1, \dots, y_{m-1})$

En suivant cette définition, il suffit de remplir ensuite une matrice de programmation dynamique de taille  $n + 1$  par  $m + 1$ .

**Q 4** Que vaut  $x \leq y$  si  $x$  est plus longue que  $y$  ?

**Q 5** Comment va être initialisée la première ligne de la matrice ?

**Q 6** Comment va être initialisée la première colonne de la matrice ?

**Q 7** Dans la matrice de programmation dynamique, où se trouve le résultat du calcul effectué ?

**Q 8** Ecrire une fonction utilisant la programmation dynamique `function inferieur (x, y : LISTE_MATRICULES) return BOOLEAN;`

**Q 9** Donner la complexité en temps et en espace.