

# Initiation à l'algorithmique

## corrigé DS

### Exercice 1.1

#### Question de cours :

*Parmi les algorithmes de tri de tableaux par insertion, par sélection et à bulle indiquez ceux qui après une exécution partielle produisent un tableau partiellement trié. C'est à dire un tableau dans lequel une partie des données occupent déjà leur place définitive dans le tri. Justifiez.*

On se place dans le cadre d'un tri croissant.

Tri sélection : oui. A chaque itération on cherche le minimum parmi les valeurs non encore triées et on le met à sa place définitive.

Tri insertion : non . Après  $i$  itérations les valeurs situées sur les  $i$  premières places dans le tableau initial sont triées entre elles. Parmi les éléments non encore triés il peut y avoir des éléments inférieurs à ceux déjà triés.

Tri à bulle : oui. A chaque itération les "petites" valeurs se déplacent vers la gauche du tableau, les "grandes" vers la droite. Le minimum parmi les valeurs non encore à leur place définitive atteint sa place définitive.

### Exercice 1.2

Soit la structure *date* suivante :

```
date = structure
    jour: entier;
    mois: entier;
    annee: entier;
finStructure
```

avec

- $1 \leq jour \leq 31$ ,
- $1 \leq mois \leq 12$ .

1. *Ecrivez une fonction d'en-tête `estInférieure` (val  $a$  : date, val  $b$  : date) : booléen retournant **vrai** si la date  $a$  est strictement antérieure à la date  $b$ , **faux** sinon.*

```
fonction EstInferieure ( val a : date, val b : date) : boolean
Debut
    retourner ((a.annee < b.annee) ou (a.annee == b.annee et
a.mois < b.mois) ou (a.annee == b.annee et
a.mois == b.mois et a.jour < b.jour))
fin.
```

2. *En utilisant un algorithme de **tri par sélection**, écrivez une fonction d'en-tête `tri` (ref  $t$  : tableau[1..NB\_DATES] de dates) : vide qui trie le tableau passé en paramètre dans l'ordre croissant.*

```

fonction MinSoustableau (ref T : tableau [1..NB_DATES] de date,
                        val Imin, Imax : entier) : entier;
    var sauv : entier;
    Debut
        sauv = Imin;
        pour i allant de Imin +1 à Imax faire :
            si EstInferieure ( T[i], T[sauv]) alors
                sauv = i;
            finsi
        finpour
        retourner (sauv)
    fin
finFonction

```

```

fonction Tri (ref T : tableau [1..NB_DATES] de date) : vide;
    var i, indice_cle : entier;
    Debut
        pour i allant de 1 à NB_DATES -1 faire
            indice_cle = MinSoustableau (T, i, NB_DATES);
            echange (T[i], T[indice_cle]);
        finpour
    fin
finFonction

```

3. Soit la nouvelle structure suivante :

```

personne = structure
    nom: chaîne de caractères;
    jour: entier;
    mois: entier;
    annee: entier;
finStructure

```

*Supposez qu'on veuille trier par dates croissantes un tableau de personnes dont les noms sont déjà triés. Ce même tri par sélection est-il approprié ? Justifiez.*

Non ce tri n'étant pas stable rien ne garantit que le tri alphabétique sera conservé après un tri selection.

### Exercice 1.3

On souhaite trier un tableau  $T$  de  $n$  booléens de façon à ce que les valeurs fausses soient rangées au début du tableau.

Par exemple si

$$T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 0 & \dots & 1 & 0 & 0 & 1 \\ \hline \end{array}$$

avec  $0 = FAUX$  et  $1 = VRAI$

Après l'application de l'algorithme de tri on aura

$$T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & \dots & 0 & 1 & \dots & 1 & 1 & 1 \\ \hline \end{array}$$

En cours d'algorithme une partie des données sont déjà triées et le tableau est organisé de la façon suivante :

$$T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline & 1 & & i & & j & & n \\ \hline 0 & 0 & 0 & ? & ? & ? & ? & 1 & 1 & 1 \\ \hline \end{array}$$

Les "?" représentent les données non encore traitées.

1. *Que représentent  $i$  et  $j$  ?*

$i$  et  $j$  sont les bornes de la zone qui contient les éléments non encore triés dans le tableau  $T$ .  $T[i]$  est le premier "?",  $T[j]$  est le dernier "?".

2. *Selon la valeur de  $T[i]$  quel traitement doit-on effectuer ?*

Deux cas possibles :

(a) si  $T[i] == 0$  on incrémente  $i$ .

(b) si  $T[i] == 1$  on échange  $T[i]$  avec  $T[j]$  on décrémente  $j$ .

3. *Quand doit-on arrêter l'algorithme ? on arrête quand  $i == j$*

4. *Ecrire l'algorithme de tri.*

```
fonction Tri2couleurs (ref T : tableau [1..N] de booléens):vide
var i, j : entier;
Debut
  i =1; j = N;
  tant que (i < j) faire
    si T[i] == 0 alors
      i = i+1;
    sinon
      echanger (T[i], T[j])
      j = j-1;
  fin tant que
fin
finfonction
```

5. *Quelle est sa complexité en temps ?* Chaque élément du tableau est étudié une seule fois avant de trouver sa place définitive donc la complexité est linéaire par rapport au nombre  $N$  d'éléments du tableau.