

## Algorithmique et programmation 3

### EXERCICES

#### 1-Complexité des programmes

##### Exercice 1-1 : complexité asymptotique du test de présence dans un tableau

La fonction de test de présence d'une valeur dans un tableau de N éléments est la suivante :

```
fonction Presence( $\downarrow$ val : Valeur ;  $\downarrow$ tab : Tableau ;  $\downarrow$ N : entier) : booléen
début
    trouve  $\leftarrow$  faux
    i  $\leftarrow$  0
    tant que non trouve et (i < N) faire
        si tab[i] = V alors
            trouve  $\leftarrow$  vrai
        sinon
            i  $\leftarrow$  i + 1
        fsi
    ftq
    retourne trouve
fin
Tableau = tableau de Valeur[N]
```

- Déterminer la taille du problème.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique (avec preuve à l'appui) du pire des cas.
- Proposer une fonction de test de présence d'une valeur dans un tableau **trié**. Répondre aux mêmes questions que pour la fonction `Presence`. On attend une complexité asymptotique strictement inférieur à celle de `Presence`.

##### Exercice 1-2 : complexité asymptotique de l'inversion d'un tableau

- Ecrire l'algorithme qui permet d'inverser les éléments d'un tableau.
- Déterminer la taille.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique (avec preuve à l'appui).

##### Exercice 1-3 : complexité asymptotique de la somme dans un tableau

- Ecrire l'algorithme qui calcule la somme de tous les éléments positifs d'un tableau.
- Déterminer la taille.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique (avec preuve à l'appui).

#### Exercice 1-4 : complexité asymptotique d'un algorithme simple

- Ecrire l'algorithme qui affiche les entiers  $N, N/2, N/4, \dots, 0$  pour un  $N$  donné.
- Déterminer la taille.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique

Remarque: on considère que la division entière ( $m/n$ ) a un coût 1.

#### Exercice 1-5 : complexité asymptotique dans le pire des cas du tri par insertion

Le tri par insertion d'un tableau de  $N$  éléments consiste, pour  $i$  allant de 1 à  $N-1$ , à insérer le  $i^{\text{ème}}$  élément à la position qu'il doit occuper dans la partie  $[0, i]$  du tableau pour que cette partie soit triée  
→ après une itération, le tableau est trié de 0 à 1 et, après  $N-1$  itérations, il l'est de 0 à  $N-1$ .

En voici **une** implémentation :

```
fonction tri(↑tab : tableau d'entiers[N]; N : entier)
début
  pour i de 1 à N-1 faire
    // On cherche la position de tab[i] dans tab[0..i]
    pos ← 0
    tant que (tab[i] > tab[pos]) faire
      pos ← pos+1
    ftq
    temp ← tab[i]

    // On décale à droite les éléments de pos à i-1
    pour j décroissant de i à pos+1 faire
      tab[j] ← tab[j-1]
    fpour
    tab[pos] ← temp
  fpour
fin
```

- Déterminer la taille du problème.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions exécutées dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique (avec preuve à l'appui).
- Comment améliorer l'algo ? Cela change-t-il la complexité asymptotique ?

#### Exercice 1-6 : complexité asymptotique du produit de deux matrices

- Ecrire l'algorithme.
- Déterminer la taille.
- Déterminer le pire des cas.
- Exprimer le nombre d'instructions dans le pire des cas en fonction de la taille.
- Donner la complexité asymptotique (avec preuve à l'appui).