

École Supérieure Polytechnique de
Dakar – Département Génie
Informatique – DIC 1

Travaux dirigés d'Algorithmique
(Analyse et conception des algorithmes)

Exercice 1 :

Soit A un tableau de n entiers ($n \geq 1$).

1. Écrire un algorithme itératif qui calcule la somme des éléments de A et prouver cet algorithme.
2. Déterminer la complexité de cet algorithme.
3. Écrire un algorithme récursif qui calcule la somme des éléments de A et prouver cet algorithme.
4. Déterminer la complexité de cet algorithme.

Exercice 2:

Soit la fonction récursive F suivante :

```
fonction F(n : entier)
  si n=0
    alors
      Retourner (2)
  sinon
    Retourner (F(n-1)+F(n-1) )
fin si
```

1. Que calcule cette fonction ? Le prouver.
2. Déterminer la complexité de la fonction F . Comment améliorer cette complexité ?
3. Écrire une fonction itérative G équivalente à F . Le prouver.
4. Déterminer la complexité de la fonction G .

Exercice 3

Soit $T[1..n]$ un tableau de n entiers ($n \geq 1$). On définit l'algorithme suivant :

Fonction Partition (A : tableau ; i, j , clé : entier) : entier

$d \leftarrow i$; $f \leftarrow j$

Répéter

Tant que ($d \leq j$) et ($T[d] \leq \text{clé}$) **faire**

$d \leftarrow d+1$

fin tant que

Tant que ($f \geq i$) et ($T[f] > \text{clé}$) **faire**

$f \leftarrow f-1$

fin tant que

Si $d < f$

Alors

Echanger $T[d]$ et $T[f]$

$d \leftarrow d+1$; $f \leftarrow f-1$

fin si

Jusqu'à $d > f$

Retourner (f)

1. Écrire et prouver les invariants de boucles correspondant aux différentes étapes de l'algorithme.
2. Quelle est la complexité dans le pire des cas de la fonction Partition ?

Exercice 4

Le tri par insertion d'un tableau $T[1..n-1]$ de n éléments consiste à réaliser un tri préliminaire du tableau $T[1..n-1]$, puis insérer le dernier élément du tableau à sa place dans le tableau, en décalant d'une position vers la droite les éléments qui le suivent.

1. Écrire une procédure *Tri_Insertion* récursive.
2. Démontrer que *Tri_Insertion* réalise le tri du tableau T par ordre croissant.
3. Déterminer la complexité du tri par insertion en nombre de comparaisons de clés.