

# Fiche TD 12 :

## Notion de complexité

Licence 1 MASS semestre 2, 2006/07

### Exercice 1 : Calcul de complexité d'algorithmes itératifs

- a - Tracer en fonction de  $n$  la complexité des six algorithmes de l'exercice 1 du TD.
- b - Quelle est la taille des données que l'on peut traiter avec chaque algorithme pour une complexité de 300 ?

### Exercice 2 : Suite de Fibonacci

- a - Reprendre l'algorithme de Fibonacci que vous avez déjà écrit. Quel est le plus grand terme que vous pouvez calculer à l'aide de cet algorithme ?
- b - Utiliser un tableau pour obtenir une complexité linéaire. Quel est le plus grand terme que vous pouvez calculer à l'aide de cet algorithme ?

### Exercice 3 : Recherche dichotomique

Consulter l'aide de l'instruction "time()" qui permet de consulter l'horloge de la machine.

Mesurer et tracer le temps d'exécution de l'algorithme de recherche dichotomique en fonction de la taille du tableau.

### Exercice 4 : Algorithmes de multiplication

<b>Algorithme</b> multiplication1(a, b : entier) : entier	<b>Algorithme</b> multiplication2(a, b : entier) : entier
<b>début</b>	<b>début</b>
<b>variable</b> res : entier	<b>variable</b> res : entier
res ← 0	res ← 0
<b>pour</b> i de 1 à a <b>faire</b>	<b>pour</b> i de 1 à b <b>faire</b>
res ← res + b	res ← res + a
<b>fin pour</b>	<b>fin pour</b>
retourner res	retourner res
<b>fin</b>	<b>fin</b>

**Algorithme** multiplication3(a, b : entier) : entier

```
début
variable S : entier
  S ← 0
  tant que a > 1 faire
    si modulo(a, 2) = 1 alors
      S ← S + b
    fin si
    a ← E[a/2]
    b ← 2b
  fin tant que
  retourner S
fin
```

Question :

- Comparer les temps d'exécution des algorithmes précédent. Est-ce en accord avec le calcul de complexité ?

## Exercice 5 : Algorithmes récursifs

- c - Ecrire un programme qui calcule la somme de  $k$  nombres successifs contenus dans une liste.
- d - Ecrire un programme qui calcule la plus grande somme de  $k$  nombres successifs contenus dans une liste.