

## NOTIONS DE COMPLEXITÉ

Matériel en grande partie dû à Michel Syska.

Lien web : [http://www-sop.inria.fr/members/Stephane.Perennes/TD2/TD2\\_Complexite.pdf](http://www-sop.inria.fr/members/Stephane.Perennes/TD2/TD2_Complexite.pdf)

### Rendu de TD

Les réponses aux exercices sont envoyées par email à [stephane.perennes@inria.fr](mailto:stephane.perennes@inria.fr) avec pour objet "Rendu TD IUT 2802", en respectant les consignes suivantes :

- un fichier `README.txt` qui décrit votre avancement : exercices terminés, en cours, difficultés éventuelles rencontrées, ...
- quand c'est demandé, fournir un document texte de nom `exercice_num.txt` avec les réponses de l'exercice numéro `num` ;
- les fichiers sources Java **commentés avec votre/vos noms en en-tête** doivent utiliser le codage UTF-8 et respecter les noms de l'énoncé ;
- chaque programme doit proposer dans la fonction `main()` des démonstrations en mode silencieux (aucune entrée/sortie interactive).

### 1 Notations asymptotiques

Les réponses à cet exercice seront données dans le fichier `exercice_1.txt`. Pour les symboles, il suffit d'écrire  $\Omega(n)$  pour  $\Omega(n)$ ,  $\Theta(n)$  pour  $\Theta(n)$ , ...

1. Application des définitions du cours *Notions de complexité* (diapositives 8 à 13)

- (a) En utilisant la définition de  $O$  montrer que  $3n^2 + 4n + 6 = O(n^2)$ .
- (b) En utilisant la définition de  $\Omega$  montrer que  $3n^2 + 4n + 6 = \Omega(n^2)$
- (c) En utilisant la définition de  $\Theta$  montrer que  $3n^2 + 4n + 6 = \Theta(n^2)$

2. Application des propriétés du cours *Notions de complexité* (diapositive 14)

Caractériser les fonctions suivantes en termes de notation  $\Theta$  :

- (a)  $2^{1024} + 5n^8 + 2087n^7$
- (b)  $1024n + 2048\sqrt{5n}$
- (c)  $2n^8 + 6789n^8 + \frac{3}{2}n^9$
- (d)  $\frac{n^2}{3} + 10^{100}n$
- (e)  $4\log(5n) + \sqrt{3n}$
- (f)  $2n \log n + 10\sqrt{n}$
- (g)  $2n \sum_{i=1}^{2n} (3i + 1) + \sum_{p=1}^n 2^p$

3. Comparer les fonctions suivantes (qui est en  $O()$  de qui ?) :

- (a)  $2^n$
- (b)  $n^n$
- (c)  $n!$

4. En utilisant la définition de  $\Theta$  montrer que  $f(n) + g(n) = \Theta(\max(f(n), g(n)))$

## 2 Complexité des algorithmes du TD1

1. Exercice 1 du TD1. Exprimer en notations  $\Omega$ ,  $\Theta$  et  $O$  le nombre de fois où s'affiche « Bonjour 1 » et « Bonjour 2 ».
2. Exercice 2 du TD1. Exprimer en notations  $\Omega$ ,  $\Theta$  et  $O$  la complexité du tri par sélection.
3. Exercice 3 du TD1. Exprimer en notations  $\Omega$ ,  $\Theta$  et  $O$  la complexité de la recherche dichotomique (aussi appelée recherche binaire).

## 3 Exercice 4 du TD1

Terminer l'exercice 4 du TD1 et écrire la complexité de chacune des trois méthodes dans les commentaires, juste à la suite des Pre et Post conditions.

## 4 Au voleur !

Un voleur cherche à dévaliser un magasin. Comme il a mal au dos, il ne peut pas porter plus qu'un poids maximum `poidsMax` mais il cherche cependant à emporter le butin de plus grande valeur possible. Le butin est choisi parmi les `Articles` du `Magasin`, pris en 0 ou 1 exemplaire.

On considère la classe `Magasin` donnée sur Moodle : `m415-skel-td2.zip`

- Pour l'instance de problème du magasin bio (constructeur par défaut de la classe `Magasin`), quelle est la valeur optimale du butin qu'il emportera pour des valeurs de poids maximum de 30, 311 et 1000?
- Compléter la méthode `private int volerAux(int poidsMax, int j)` de la classe `Magasin` qui calcule la valeur maximale possible du butin par une recherche de type « force brute ».
- Donner une 2<sup>e</sup> version de cette méthode qui renvoie la liste des articles retenus en plus de la valeur pour le butin optimal. Utiliser la classe `SacADos` qui implémente une liste et non un tableau.