

## Algorithmique (AL5)

### TD n° 1 : Complexité, Graphes

#### Comparaison asymptotique

##### Exercice 1 : Ordres de grandeur

Est-ce que les appartenances ci-dessous sont correctes ? Justifier vos réponses.

- |                                        |                                                 |
|----------------------------------------|-------------------------------------------------|
| 1. $3n^2 + 4n - 6 \in O(n^2)$ ?        | 9. $3 + 5 \cdot  \sin(n)  \in \Theta(1)$ ?      |
| 2. $3n^2 + 4n - 6 \in O(n^5)$ ?        | 10. $2n + 3 \in \Theta(n)$ ?                    |
| 3. $3n^2 + 4n - 6 \in \Theta(n^2)$ ?   | 11. $3^n \in O(2^n)$ ?                          |
| 4. $3n^2 - 4n - 6 \in \Theta(n^4)$ ?   | 12. $(n + 1)! \in O(n!)$ ?                      |
| 5. $3n^3 - 4n^2 - 6 \in \Theta(n^3)$ ? | 13. $n! \in O(n^n)$ ?                           |
| 6. $3n^2 + 2^n \in \Theta(2^n)$ ?      | 14. $n^n \in O(n!)$ ?                           |
| 7. $3n^2 + 2^{3n+2} \in \Theta(2^n)$ ? | 15. $n^n + 2^n + n^{10} + n! \in \Theta(n^n)$ ? |
| 8. $3n^2 + 2^{3n^2} \in O(2^{n^3})$ ?  |                                                 |

##### Exercice 2 : Complexité d'algorithmes

- Quelle est la complexité (en temps dans le pire cas) de l'algorithme **Algo1** qui manipule un graphe orienté  $G = (S, A)$  ? On distinguera le cas où  $G$  est représenté sous forme de liste d'adjacence ou sous forme matricielle. On donnera un ordre de grandeur de la complexité sous forme de  $\Theta$ .

```

Algo1 (G) :
// G = (S,A)
Pour x in S:
    Pour (x,y) in A :
        print("(x -> y)")

```

- Que fait l'algorithme **Algo2** et quelle est sa complexité ? On supposera que le graphe  $G = (S, A, w)$  est un graphe orienté valué avec  $w : A \rightarrow \mathbb{N}$  et que pour tout sommet, il existe au moins un arc qui part de  $x$ .

```

Algo2 (G) :
// G = (S,A,w)
V est un tableau d'entiers de taille |S|
Trier les arcs par poids w croissant
Pour x dans S:
    V[x] = undef
    Pour (x,y) in A [pris dans l'ordre...]
        Si V[x] == undef Alors V[x] = w(x,y)
Retourner V

```

- Proposer un algorithme plus efficace pour remplacer **Algo2**.

**Exercice 3 : Modélisation**

Tom a une chèvre, un loup et un chou. Il veut traverser une rivière, avec une petite barque où il ne peut emporter qu'un seul passager (loup, chèvre ou chou) en plus de lui. Il ne peut pas laisser la chèvre seule avec le loup car il la mangerait, ni la chèvre seule avec le chou... Trouver une solution en modélisant le problème sous la forme d'un graphe non orienté.

**Décrassage algorithmique****Exercice 4 :**

Un tablette de chocolat est composée de  $n \times m$  carreaux. On souhaite séparer tous les carreaux, et pour cela la seule opération dont on dispose est de prendre un morceau de chocolat et de le couper en deux verticalement ou horizontalement (en suivant une ligne ou une colonne).

1. Proposer un algorithme pour séparer tous les carreaux. Quelle est sa complexité? Appliquer le sur une tablette  $2 \times 5$ .
2. Quel est le nombre minimal d'opérations permettant de séparer tous les carreaux (quel que soit l'algorithme)? Votre algorithme est-il optimal?

**Exercice 5 : Des pièces et une balance**

1. Vous avez 12 pièces, toutes de même poids exceptée une qui est légèrement plus légère que les autres. Vous disposez d'une balance Roberval à *deux plateaux*. Trouver la pièce "lourde" en 3 pesées seulement. Même problème avec 27 pièces.
2. Généraliser votre algorithme à tout  $n$  et estimer le nombre de pesées nécessaires. Justifier la correction de votre algorithme.
3. Deux extensions :
  - (a) Que se passe-t-il si la pièce recherchée est plus lourde ou plus légère?
  - (b) Que se passe-t-il si il y a deux pièces plus lourdes?

