

Nous avons vu jusqu'à présent comment déterminer la complexité d'un algorithme que ce soit une des complexités classiques qui ont leur propre nom ou des complexité du type "proportionnelle à ...". Il existe une notation simple qui évite de répéter à chaque fois la phrase "proportionnelle à ...". Par exemple si la complexité d'un algorithme est proportionnelle à  $N^2$  alors on notera que la complexité de l'algorithme est un  $O(N^2)$  qu'on lira "grand Ô de  $N$  au carré".

Pour les complexités classiques, on a donc :

- Un algorithme cubique à une complexité en  $O(N^3)$ .
- Un algorithme quadratique à une complexité en  $O(N^2)$ .
- Un algorithme linéaire à une complexité en  $O(N)$ .
- Un algorithme constant à une complexité en  $O(1)$ .

Ne pas oublier que cette notation revient à dire "si  $N$  est assez grand alors la complexité est proportionnelle à ...". En particulier, si on a un algorithme en  $O(N^3)$  et un autre en  $O(N^2)$  alors exécuter ces deux algorithmes l'un après l'autre donne un temps de calcul en  $O(N^3)$  car le  $O(N^2)$  est négligeable par rapport au  $(N^3)$ .

Il existe une véritable définition derrière cette notation, mais nous ne verrons pas cela tout de suite, nous aurons l'occasion de présenter les choses plus formellement dans le chapitre sur la "Complexité avancée".