



Complexité algorithmique (Introduction)



LE CREN Anthony

L'analyse de la complexité d'un algorithme consiste en l'étude de la quantité de ressources (par exemple de temps ou d'espace mémoire) nécessaire à l'exécution de cet algorithme. (Wikipédia)

Le terme de « complexité » est trompeur parce qu'on ne parle pas d'une difficulté de compréhension d'un algorithme, mais le but est d'évaluer son **efficacité**, en se demandant : "est-ce que l'exécution de cet algorithme est rapide ?".



Depuis de nombreuses années, les ordinateurs sont de plus en plus rapides et puissants, avec plus de mémoire. Mais les problèmes que l'on aborde ont souvent un **volume de données importantes à traiter**, cela implique donc plus de temps de traitement et de ressources.



Exemple : pour rechercher un numéro de téléphone dans l'annuaire, l'utilisateur ne devra pas attendre 2 heures.

Attention, changer d'ordinateur pour un plus puissant ne permet pas d'améliorer l'efficacité d'un algorithme. En effet si le volume de données à traiter est de plusieurs milliards de giga-octets, cela ne changera pas grand-chose et l'utilisateur devra peut-être attendre des siècles avant d'avoir l'information voulue.



En fonction de n :

- Combien d'opérations cela va prendre ?
- En combien de temps ?

Remarque :

Les entrées peuvent être : un fichier de données, un tableau de valeurs en mémoire vive (RAM), une image, un son etc...

Objectif :

Vous allez maintenant mettre en œuvre 3 algorithmes différents et mesurer le temps de calcul pour chacun des 3 programmes de manière expérimentale.

En fonction du volume de données n à traiter, vous tracerez une courbe (temps de calcul = $f(n)$) avec un tableur et déterminerez la complexité du programme en vous aidant du document ressource.

Pour finir, vous complétez 3 fiches en concluant sur la complexité.

Algorithme N°1 : transformer une image en nuances de gris**Algorithme N°2 : Tri croissant de valeurs entières par sélection**

Avant tri :

[128, 291, 9, 57, 107, 330, 5]

Après tri croissant :

[5, 9, 57, 107, 128, 291, 330]

Algorithme N°3 : Recherche dichotomique dans un tableau trié de valeurs entières

Indice	0	1	2	3	4	5	6
Valeur	1	5	12	45	201	452	987

Valeur 45 à l'indice 3 dans le tableau