

Chapitre 9

Comparaison d'algorithmes

1 Qu'est-ce qu'un algorithme ?

Définition

Un algorithme est un ensemble d'étapes qui permettent d'accomplir une tâche.

Un algorithme résout un problème

- proposer une solution juste/exacte à partir des données en entrée
peu importe l'ordre de ces données
- utiliser les ressources de calcul de manière efficace

2 Temps d'exécution d'un algorithme

Important

Le **temps d'exécution** de l'algorithme est une première mesure de l'efficacité de l'algorithme

Le temps d'exécution dépend de facteurs **internes** et **externes** à l'algorithme :

- **externes** : rapidité de l'ordinateur, langage de programmation utilisé, compilateur / interpréteur utilisé, ce qui tourne en même temps sur l'ordinateur, de la taille des entrées à traiter
- **internes** : comment le programme a été écrit

Exercice 25 (Exemple)

On considère 2 algorithmes dont l'exécution prend un certain nombre de cycles processeur en fonction de la taille n des entrées (n est le nombre d'éléments à traiter, comme le nombre d'éléments d'un tableau) :

nb de cycles processeur	ordre de grandeur qd n grand	taux de croissance retenu
$50n + 125$		
$20n^3 + 100n^2 + 212$		

nb de cycles processeur	ordre de grandeur qd n grand	taux de croissance retenu
$50n + 125$	$50n$	n
$20n^3 + 100n^2 + 212$	$20n^3$	n^3

Les facteurs 50 et 20 ont une importance mais dépendent tellement des paramètres externes à l'algorithme qu'on ne peut pas les comparer d'un ordinateur à un autre.

Soit A et B 2 implémentations différentes du même algorithme. Il se peut tout à fait que l'algo A soit plus rapide que l'algo B sur une certaine combinaison de machine, langage de programmation, compilateur / interpréteur alors qu'une autre combinaison favorise l'algo B . Donc on ne tient pas compte de ces facteurs, on ne considère que l'évolution du taux de croissance en fonction de la taille des entrées n !

3 Logarithme en base 2

Définition du logarithme en base 2

On note $\lg n$ le logarithme de n en base 2 qui est l'inverse de la fonction exponentielle 2^n :

$$n = 2^x \quad \Leftrightarrow \quad x = \lg n$$

- $\log x$ est le logarithme décimal, fonction réciproque de la fonction exponentielle 10^x
- $\ln x$ est le logarithme népérien, fonction réciproque de la fonction exponentielle e^x
- vous n'avez pas cette fonction directement sur votre calculatrice, mais vous avez le logarithme népérien $\ln x$ qui permet de calculer le logarithme en base 2 par la relation

$$\lg x = \frac{\ln x}{\ln 2}$$

- pour calculer un tel logarithme avec python : `import math`, puis `math.log(x,2)` pour être en base 2 !

Exercice 26 (Application)

x	$\lg x$
$1024 = 2^{10}$	10
$1048576 = 2^{20}$	20

4 Classement des taux de croissance

Il faut savoir comment se comporte les taux de croissance qu'on rencontre souvent les uns par rapport aux autres. Par ordre décroissant, **du plus rapide au plus lent lorsque la taille n des entrées devient de plus en plus grand**, on a

nb de cycles processeur	taux de croissance retenu	commentaire
125	1	temps constant
$4 * \lg n + 212$	$\lg n$	logarithmique
$12 * n + 123$	n	n
$4 * n * \lg n + 212$	$n \lg n$	$n \lg n$
$3 * n^2 + 123 * n + 32$	n^2	n^2
$3 * n^3 + 14 * n^2 + 3$	n^3	n^3
$3 * 2^n + 123 * n^8 + 32$	2^n	exponentiel

Nous allons voir différents algorithmes cette année et on les classera suivant leur performance.