

## Définition d'algorithme

Le différence entre un algorithme et un *programme* est souvent une question de niveau de détail. Un algorithme est souvent exprimé avec une notation indépendante de tout *langage de programmation* alors qu'un programme est écrit dans un langage de programmation particulier.

Une autre différence entre algorithme et programme est que l'exécution d'un algorithme doit toujours se *terminer* avec un résultat, alors que celle d'un programme peut conduire à une *boucle infinie* (ne jamais s'arrêter).

Un algorithme est donc une méthode pour résoudre un problème particulier dont on est sûr qu'elle trouve toujours une réponse en un temps d'exécution fini.

Exemple : Pour déterminer si un entier  $n$  est *premier* (à savoir qu'il ne contient pas de facteur autre que  $1$  et  $n$ ), l'algorithme suivant peut être utilisé :

Pour chaque entier  $i$ ,  $2 \leq i < n$ , vérifier si  $i$  est un facteur de  $n$  (en divisant  $n$  par  $i$  puis en vérifiant si le résultat est entier). Si c'est le cas, arrêter avec la réponse ``non". Si aucune valeur  $i$  n'est facteur de  $n$ , alors arrêter avec la réponse ``oui".

Il y a dans cette définition (informelle) d'algorithme un problème majeur. Prenons l'exemple des nombres premiers. Pourquoi peut-on supposer que l'on sait comment diviser  $n$  par  $i$  et vérifier que le résultat est un entier ? Ne faudrait-il pas avoir un algorithme pour ça aussi ? Et si jamais on trouve un algorithme pour cette division, ne faudrait-il pas trouver un algorithme pour chaque étape de celle-ci ? Et ainsi de suite. Ou, au contraire, pourquoi ne peut-on pas simplement supposer que ``vérifier si un entier est premier" est une opération élémentaire ? Pourquoi est-il nécessaire de trouver un algorithme pour cette opération ?

En d'autres termes : comment savoir si une opération est élémentaire (et ne nécessite par conséquent aucun algorithme) ?

La réponse est qu'une opération est élémentaire lorsque un ordinateur peut l'exécuter *très rapidement*, en réalité en un nombre relativement faible de *cycles d'horloge*.

Le lecteur de ce livre ne sait pas forcément déterminer si une opération est élémentaire au sens du paragraphe précédent. Heureusement cela ne sera pas nécessaire. La notion de *type abstrait* examinée dans la section suivante nous permettra de savoir si une opération est élémentaire dans le sens ci-dessus.

---

[Next](#) [Up](#) [Previous](#) [Contents](#) [Index](#)

**suivant:** [Type abstrait](#) **monter:** [La notion d'algorithme](#) **précédent:** [La notion d'algorithme](#) [Table des matières](#) [Index](#)