

## Algorithmes récursifs : quelques clarifications

Il n'est généralement pas facile de lire un algorithme récursif, et encore moins de prédire quelle sera la sortie d'un tel algorithme. Ceci est dû au fait que la récursivité permet de décrire en très peu de lignes un grand nombre d'opérations. Du coup, on est en droit de se demander comment une machine procède à l'exécution d'un algorithme récursif, car tout ceci paraît peut-être un peu "magique" au premier abord...

- Tout d'abord, il convient de se poser la question de ce qu'il se passe lors de l'exécution d'un algorithme simple, disons par exemple l'algorithme **algo** avec pour donnée d'entrée un nombre entier  $n \geq 1$  : lorsque l'instruction **algo**( $n$ ) est exécutée,

1. la machine va chercher dans sa mémoire le code associé à l'identifiant **algo** (celui-ci y est enregistré à un emplacement précis sous la forme d'une suite de 0 et de 1) ;
2. et elle va également chercher à un autre emplacement de sa mémoire la valeur de la variable  $n$ , qui y est enregistrée sous la forme d'une (autre) suite de 0 et de 1.

Puis la machine exécute le code (suivant le langage de programmation utilisé, qui permet d'interpréter la liste de 0 et de 1 enregistrée dans la mémoire) avec  $n$  en donnée d'entrée (pour laquelle un nouvel espace mémoire est alloué lors de l'exécution de l'algorithme), et sort un résultat. Voilà pour le cas simple.

- Quel se passe-t-il maintenant si à l'intérieur de l'algorithme **algo**( $n$ ), un appel à **algo**( $n - 1$ ) est effectué ? On pourrait a priori avoir l'impression d'un serpent qui se mord la queue, mais il n'en est rien. A ce moment,

1. la machine va à nouveau chercher dans sa mémoire le code associé à l'identifiant **algo** (au même emplacement que précédemment) ;
2. et elle va également chercher la valeur de la variable  $n$ , qu'elle décrémente de 1 et alloue cette nouvelle donnée d'entrée  $n - 1$  à un autre espace mémoire pour l'exécution de l'algorithme.

Puis la machine exécute le code avec cette fois  $n - 1$  en donnée d'entrée et sort un résultat. A noter que l'espace mémoire alloué à la nouvelle donnée d'entrée  $n - 1$  n'est bien sûr pas le même que celui alloué à la variable  $n$  de l'algorithme de départ ! (sinon, on aurait effectivement un serpent qui se mord la queue)

- Et lors de l'exécution de ce sous-algorithme **algo**( $n - 1$ ), ce même phénomène va se répéter... jusqu'à ce que finalement la condition de terminaison (typiquement,  $n = 1$ ) soit atteinte, ce qui terminera l'exécution de l'algorithme.