

## R1.01 - Initiation au développement

**Date:** 22 Nov 2023 - 23 Nov 2022

**Sujet:** Introduction à la récursivité

**Instructeur:** Joseph AZAR - [joseph.azar@univ-fcomte.fr](mailto:joseph.azar@univ-fcomte.fr)



Image réursive d'une femme lisant un journal

## Introduction:

La récursion est une technique qui conduit à des solutions élégantes à des problèmes difficiles à programmer à l'aide de boucles simples.

Supposons que vous souhaitiez rechercher tous les fichiers d'un répertoire contenant un mot particulier. Comment pouvez-vous résoudre ce problème?

Il existe plusieurs façons de procéder. Une solution intuitive et efficace consiste à utiliser la récursion en recherchant les fichiers dans les sous-répertoires de manière récursive.

Utiliser la récursion, c'est programmer à l'aide de **méthodes récursives**, c'est-à-dire utiliser **des méthodes qui s'invoquent elles-mêmes**. La récursivité est **une technique de programmation** utile. Dans certains cas, cela vous permet de développer une solution naturelle, directe et simple à un problème autrement difficile.

# Étude de cas : Calcul des factorielles

Une méthode récursive est une méthode qui s'invoque directement ou indirectement.

La factorielle d'un nombre  $n$  peut être définie récursivement comme suit :

$$0! = 1;$$

$$n! = n \times (n - 1)!; n > 0$$

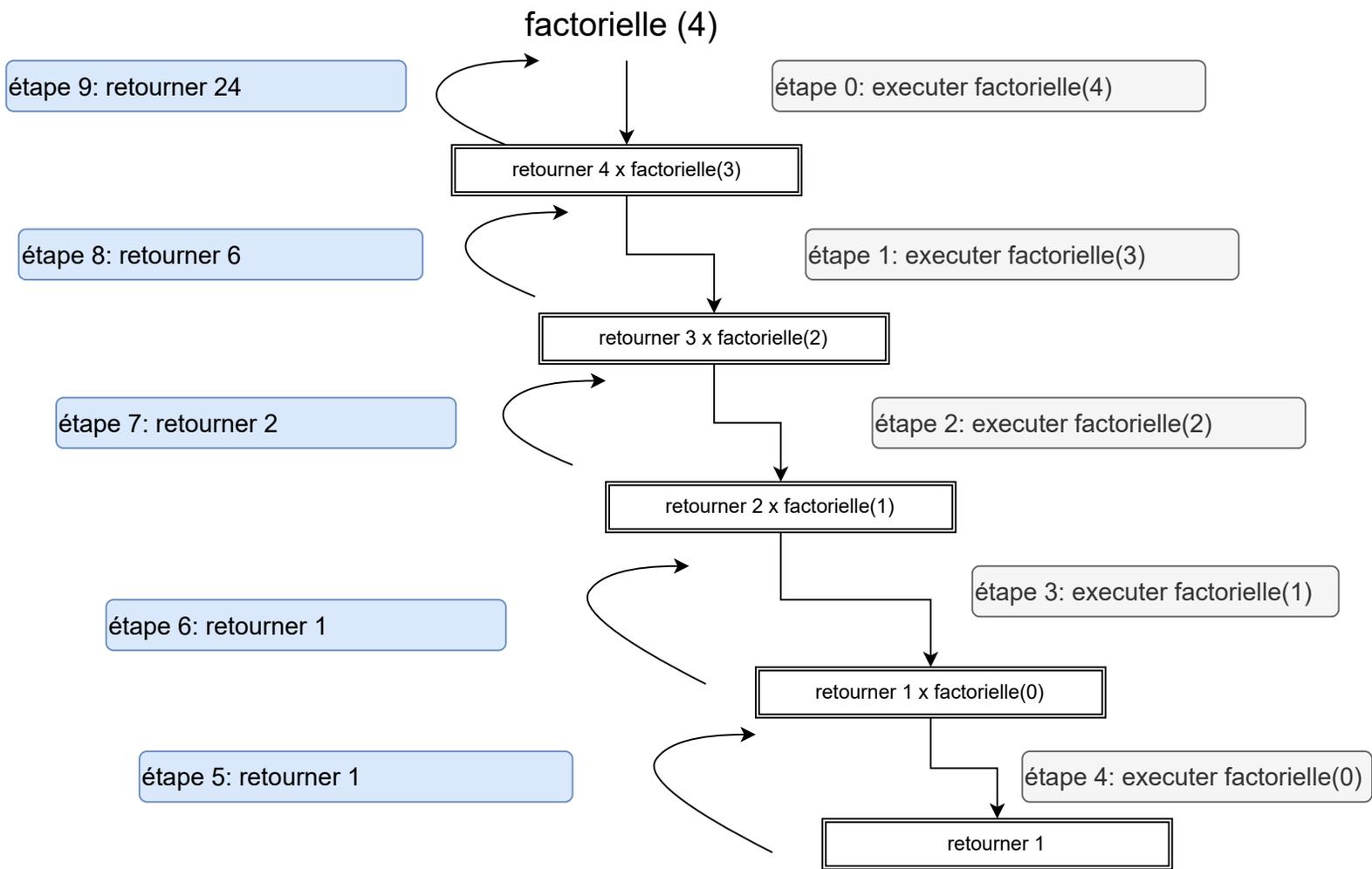
Si vous êtes confronté à un problème qui, selon vous, pourrait se prêter à une solution récursive, la première étape pour trouver une telle solution consiste à comprendre **comment le problème peut être divisé en sous-problèmes successivement plus petits.**

Chaque sous-problème doit **résoudre le même type de problème que le problème d'origine**, mais il doit résoudre son problème en utilisant un ensemble de données plus petit que celui du problème précédent.

L'étape suivante consiste à identifier une condition, appelée **condition d'arrêt**, qui indique que le problème actuel est suffisamment petit pour pouvoir être résolu sans avoir à subdiviser davantage le problème.

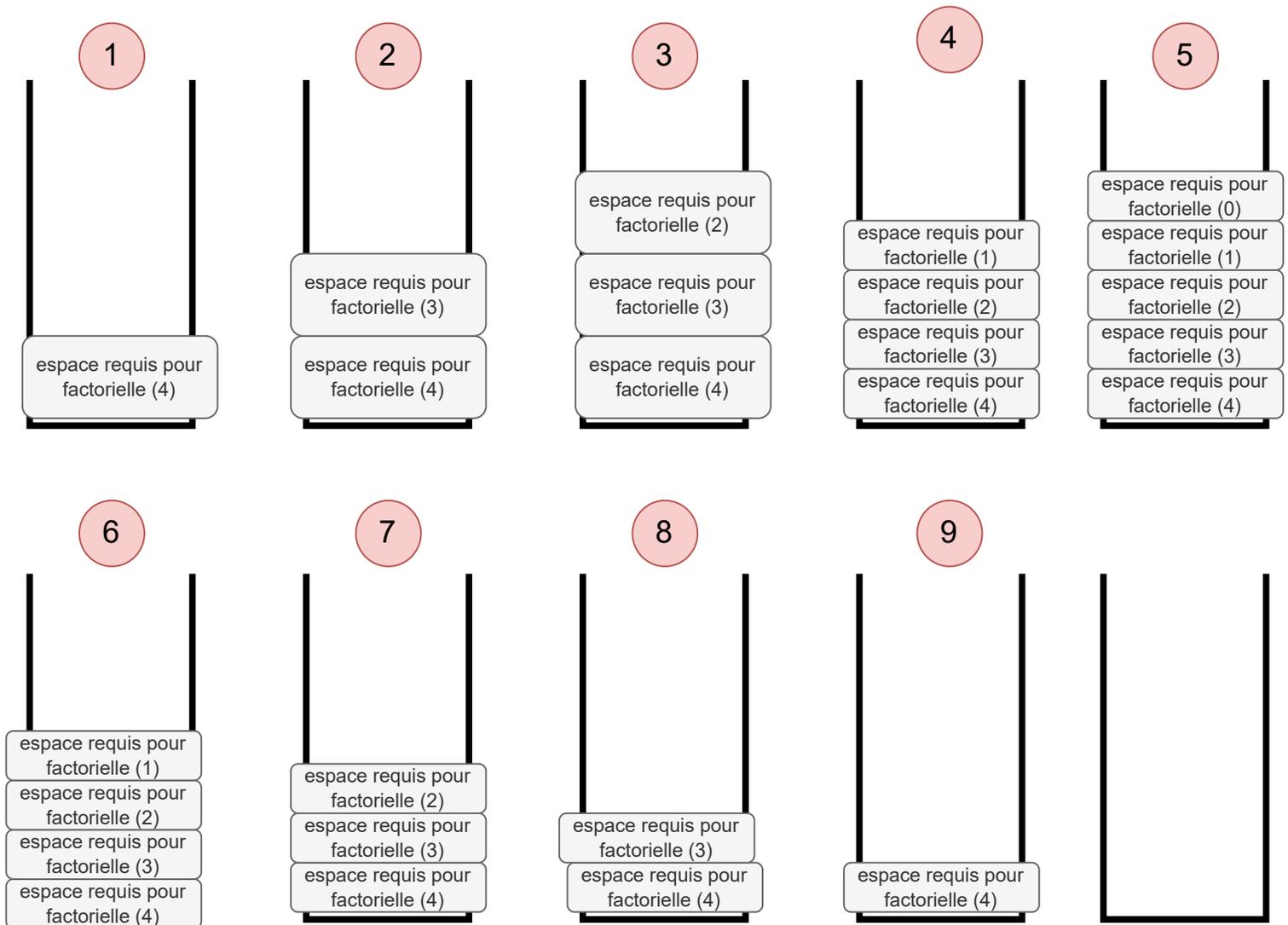


# Étude de cas : Calcul des factorielles



# Étude de cas : Calcul des factorielles

Lorsque `factorial(4)` est en cours d'exécution, la méthode factorielle est appelée de manière réursive, provoquant une modification dynamique de l'espace de la pile.



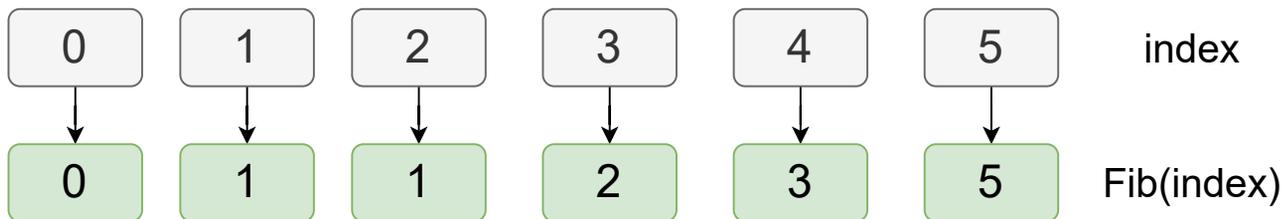
# Étude de cas : Calcul des nombres de Fibonacci

La série de Fibonacci commence par 0 et 1, et chaque nombre suivant est la somme des deux précédents. La série peut être définie récursivement comme

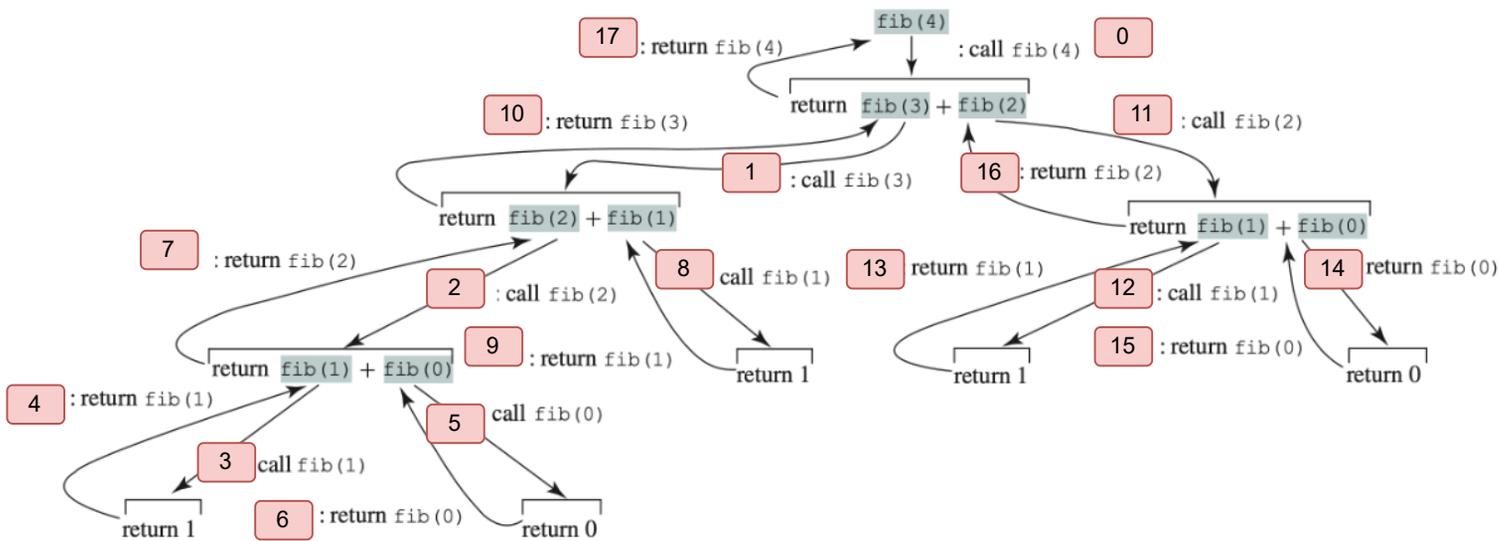
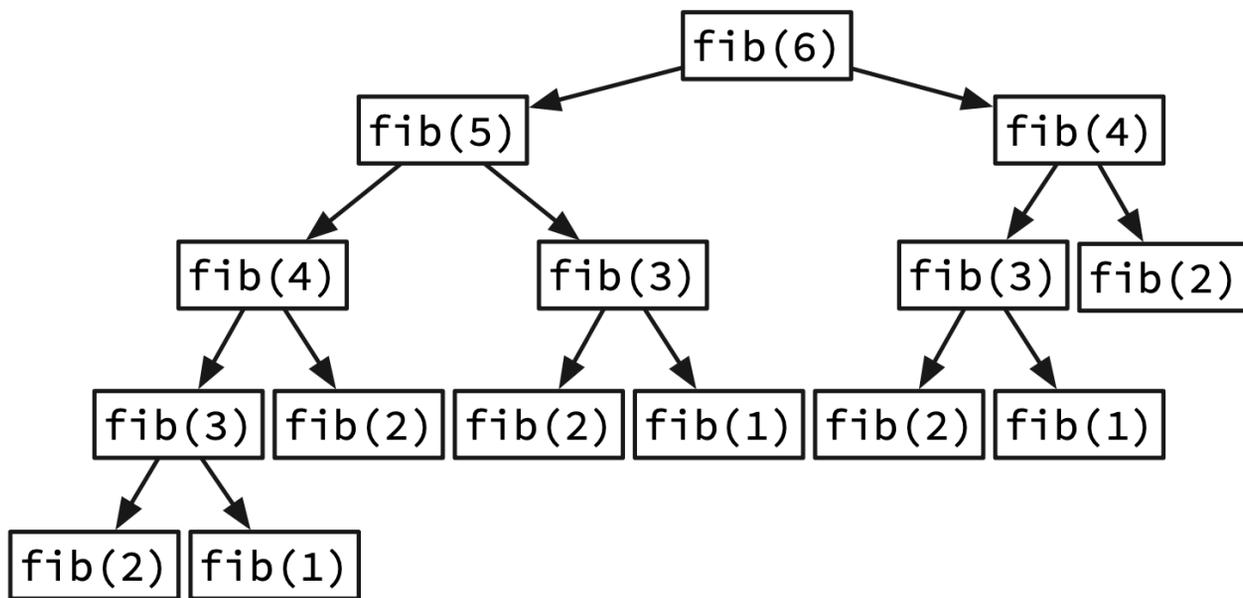
$\text{fib}(0) = 0;$

$\text{fib}(1) = 1;$

$\text{fib}(\text{index}) = \text{fib}(\text{index} - 2) + \text{fib}(\text{index} - 1); \text{index} \geq 2$

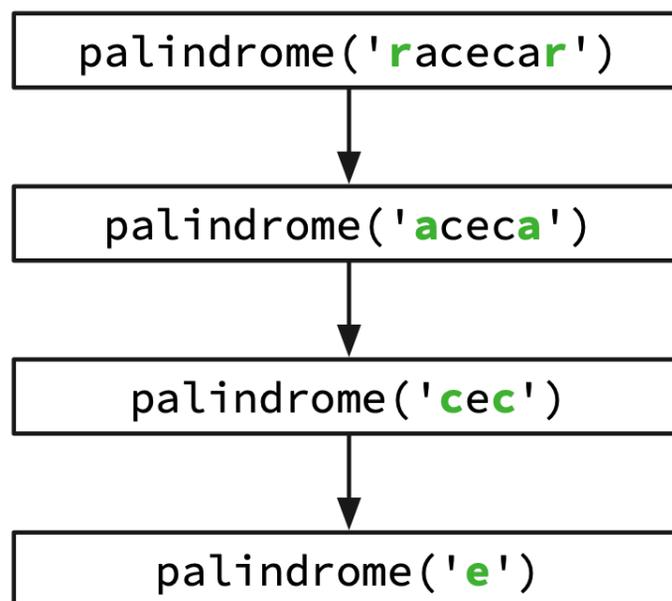


# Étude de cas : Calcul des nombres de Fibonacci



# Étude de cas : Palindromes

Une chaîne est un palindrome si elle lit la même chose vers l'avant et vers l'arrière. Les mots « mom », « dad » et « noon », par exemple, sont tous des palindromes.



## Exercice:

Résoudre le problème du palindrome en utilisant les deux stratégies : itérative et récursive

	Palindrome?
moon	false
noon	true
a	true
aba	true
ab	false