Chap 03. Récursivité

Livre p 3 – Chap 1 Récursivité
Livre p 221 – Chap 13 Diviser pour régner

1. Algorithmes récursifs

a. Introduction

La récursivité est un principe qui décrit une propriété d'objets qui se définissent à partir d'eux même.
On retrouve ce concept dans certaines œuvres qui utilisent une mise en abyme :

- les poupées russes
- les objets ou figures fractales (chou romanesco, flocon de Von Koch, tapis de Sierpinski, …)
- des tableaux, dessins ou photos (Salvador Dali, Maurits Cornelis Escher, …)
- des affiches publicitaires (la Vache qui rit, …)
- des films de Science-Fiction (Inception, TeneT, …)

Remarque : Certains sigles utilisent la récursivité par jeu et n'ont donc pas de définition précise comme :

- PHP : **P**HP **H**ypertext **P**reprocessor
- GNU : **G**NU's **N**ot **U**nix

b. Fonctions récursives

Une fonction récursive est une fonction qui s'appelle elle-même.
Attention : Une fonction récursive doit toujours comporter un ou plusieurs cas de base qui permettent de mettre fin aux appels récursifs, sinon la fonction se répète à l'infini !
Remarque : Par défaut, en python, le nombre d'appels récursifs d'une même fonction est limité à 1 000, on peut augmenter cette valeur en utilisant la commande `sys.setrecursionlimite(a)` en remplaçant `a` par la valeur désirée pour cette limite et en ayant importé la bibliothèque `sys` par « `import sys` » précédemment.

c. Spécialité Mathématiques

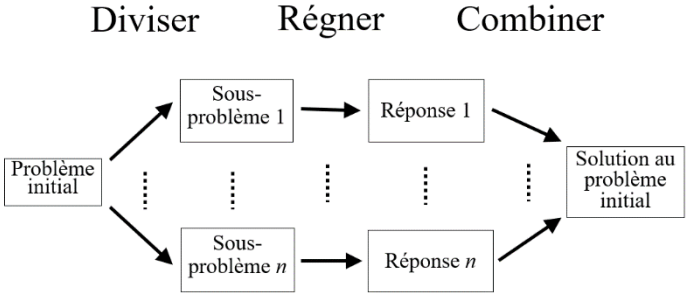
Les suites définies par une relation de récurrence du type : $u_0 = a$ et pour tout entier n, $u_{n+1} = f(u_n)$ est un exemple classique de récursivité.
Attention : Les récurrences doubles comme la suite de Fibonacci ou les coefficients binomiaux génèrent des fonctions récursives explosives, il faut alors utiliser la mémoïsation ou la programmation dynamique.

2. Méthode « diviser pour régner »

a. Principe

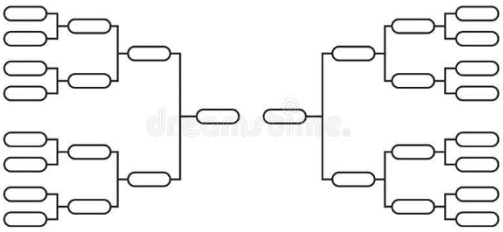
La méthode « diviser pour régner » (« divide and conquer » en anglais) se divise en trois étapes :

- **Diviser** : On découpe le problème initial en sous-problème(s) plus simple(s) pour un gain en temps, en mémoire ou en complexité.
- **Régner** : On résout le(s) sous-problème(s)
- **Combiner** : On trouve une solution au problème initial à partir des réponses obtenues au(x) sous-problème(s).



b. Exemple du tournoi sportif

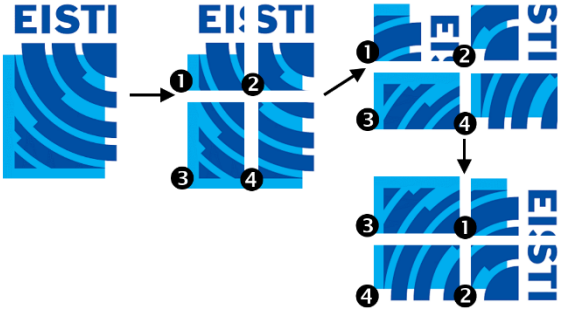
Un exemple très simple de la méthode « diviser pour régner » est la réalisation d'un tournoi sportif dans lequel on cherche la meilleure parmi une liste d'équipe donnée : On peut simplement diviser la liste des équipes en deux, déterminer la meilleure équipe de chaque demi-liste et faire s'affronter les deux meilleures équipes obtenues pour savoir quelle est la meilleure !



c. Exemple de la rotation d'un quart de tour d'une image

Un exemple classique de la méthode « diviser pour régner » en traitement d'images est la rotation d'une image. Pour faire tourner d'un quart de tour, dans le sens horaire, une image représentée par une matrice de pixels, on peut faire tourner d'un quart de tour chaque quart de l'image puis les réajuster correctement…

Exemple : https://www.youtube.com/watch?v=OXo-uzzD4Js



3. Le tri fusion (ou tri dichotomique)

a. Algorithme

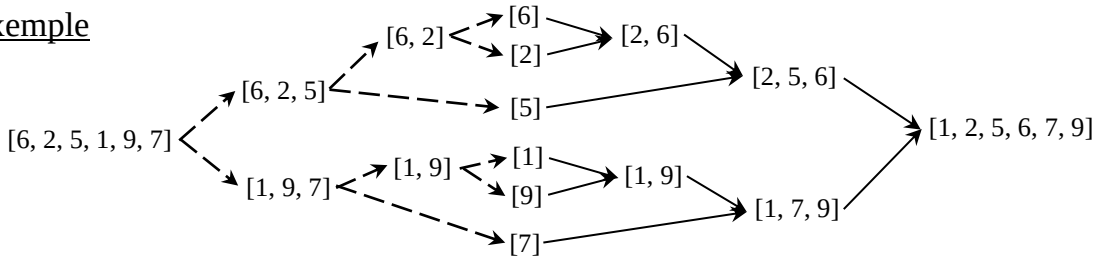
Pour trier une liste, on peut la diviser en deux sous-listes, les trier, de manière récursive, puis les fusionner. Il y a alors deux fonctions à créer :

fusion(liste_triee_1, liste_triee_2) : Elle prend en argument deux listes triées et renvoie la fusion de ces deux listes sous la forme d'une liste triée, son coût est linéaire.

triFusion(liste_a_tier) : C'est une fonction récursive qui découpe la liste à trier en deux sous-listes (tant qu'elle contient au moins 2 éléments) pour les trier puis les fusionner.

Remarque : Le coût total du tri fusion est en O(n log(n))

b. Exemple



4. Le tri rapide : QuickSort

a. Algorithme

Pour trier une liste, on peut prendre son premier élément, appelé pivot, et créer deux sous-listes :

- liste_inf : contenant tous les éléments inférieurs au pivot
- liste_sup : contenant tous les éléments supérieurs au pivot

On trie ensuite les deux sous-listes de la même manière, de façon récursive.

Puis on renvoie la liste : liste_inf_triee + [pivot] + liste_sup_triee.

Remarque : Le coût moyen du tri rapide est en O(n log(n))

b. Exemple