



Interrogation 7

Récurtivité

Correction

Exercice 1 :

Donner les syntaxes précises suivantes avec rédaction éventuelle :

1. Définition de la méthode "diviser pour régner" (les étapes et leurs définitions).

La méthode "diviser pour régner" se fait en 3 étapes : diviser (division du problèmes en plusieurs sous-problèmes de plus petites tailles) ; régner (la résolution du problème de taille minimale) ; assembler (assemblage des solutions des problèmes de tailles minimales pour reconstruire la solution au problème initiale).

2. Définition d'une fonction récursive et d'une fonction itérative.

Une fonction récursive est une fonction qui s'auto-appelle (ou qui utilise des fonctions qui l'appellent). Une fonction itérative est une fonction qui n'est pas récursive.

Exercice 2 :

La multiplication de deux entiers x et y correspond à l'addition successive de l'un, dont le nombre d'addition est fixé par l'autre. Autrement dit $x \times y = \underbrace{y + y + \dots + y}_x$ fois.

1. Définir une fonction récursive `mult(x:int, y:int) -> int` qui effectue le produit $x \times y$.

```
1 def mult(x:int, y:int) -> int :  
2     if x == 0 :  
3         return 0  
4     else :  
5         return y+mult(x-1, y)
```

2. Effectuer un jeu de tests pour $x = 5$ et $y = 3$.

Avec $x = 5$ et $y = 3$, on a le jeu de tests suivant :

x	y	Retour
5	3	3+mult(4,3)
4	3	3+mult(3,3)
3	3	3+mult(2,3)
2	3	3+mult(1,3)
1	3	3+mult(0,3)
0	3	0
1	3	3
2	3	6
3	3	9
4	3	12
5	3	15

3. Faire l'étude de la complexité de votre fonction.

On pose $C(x)$ le nombre d'opérations fait par l'appel de `mult(x,y)` pour tout $y \in \mathbb{N}$. Si $x = 0$, alors $C(0) = 2$ (une comparaison et le renvoi). Si $x \geq 1$, alors $C(x) = 1 + \max(1, 2 + C(x - 1)) = 3 + C(x - 1)$. Donc la suite $(C(x))_{x \in \mathbb{N}}$ est une suite arithmétique de raison 3. Donc $\forall x \in \mathbb{N}, C(x) = C(0) + 3x = 2 + 3x$.

Et donc, la fonction `mult` est une fonction de complexité linéaire $O(x)$.

4. Justifier de la terminaison de votre fonction.

La valeur de la variable x est une suite d'entier strictement décroissante. Et donc la valeur de la variable x est un variant de boucle. Donc l'algorithme se termine bien.