

TD 3: Algorithmique fonctionnelle

Récurtivité

Nga Nguyen - Stefan Bornhofen - Peio Loubière

1 Division entière

1.1 Quotient

Écrire une fonction récursive qui calcule le quotient de la division entière de deux nombres strictement positifs (sans utiliser l'opérateur div).

1.2 Reste

Écrire une fonction récursive qui calcule le reste de la division entière de deux nombres strictement positifs (sans utiliser l'opérateur mod).

2 Binôme $C(n,p)$

On sait que les coefficients du binôme $C(n,p)$, utilisés dans le calcul de $(a+b)^n$, sont définis par : $C(n,p) = \frac{n!}{p!(n-p)!}$ et qu'ils vérifient la propriété : $C(n,p) = C(n-1,p-1) + C(n-1,p)$.
Exemple : $(a+b)^3 = C(3,0) * a^3 + C(3,1) * a^2b + C(3,2) * ab^2 + C(3,3)b^3$

Écrire une fonction récursive qui calcule $C(n,p)$.

3 Chaîne de caractères

3.1 Palindrome

Un palindrome est une chaîne de caractères qui se lit de gauche à droite ou de droite à gauche, comme " radar ", " elle ", " été ", " esope reste ici et se repose ", si l'on fait abstraction des blancs. Écrire un algorithme récursif qui permet de vérifier si une chaîne de caractères est un palindrome ou non.

3.2 Miroir

La fonction miroir donne l'inverse d'une chaîne de caractères, c'est-à-dire la chaîne formée des mêmes caractères mais dans l'ordre inverse, par exemple miroir("séminaire") = "erianimés". Écrire un algorithme récursif qui permet de réaliser cette fonction.

4 Puissance

Soit une fonction récursive **puissance** qui prend en paramètre un réel n et un entier k et renvoie le calcul de n^k .

Question 1 : Écrire la fonction récursive **puissance** qui calcule n^k de façon naïve en considérant que $n^k = n * n^{k-1}$ et que $n^0 = 1$.

Question 2 : Transformer la version précédente en récursivité terminale.

Question 3 : Il existe une autre méthode de calcul de n^k plus efficace : la méthode par dichotomie. Elle repose sur le fait que si k est pair alors $n^k = (n * n)^{k/2}$ et $n^k = n * n^{k-1}$ sinon.

Écrire la fonction récursive **puissance-dic** qui calcule n^k par dichotomie.

Question 4 : Transformer la version précédente en récursivité terminale.