RÉCURSIVITÉ

Tous les algorithmes et fonctions dont on demande l'écriture doivent être récursifs. Les algorithmes seront décrits si possible en code CAML.

1) [Coefficients du binôme] Décrivez deux algorithmes différents de paramètres deux entiers naturels n, k qui calculent le coefficient du binôme $\binom{n}{k}$ sans utiliser la factorielle (avec la définition générale : nombre de parties à k éléments dans un ensemble à n éléments).

2) [Somme des chiffres d'un entier] Décrivez un algorithme de paramètre un entier naturel non nul qui calcule la somme de ses chiffres (en base 10).

3) On considère un sac contenant b boules blanches et n boules noires. On effectue des tirages dans le sac :
— si on tire une boule noire, on la met de côté ;
— si on tire une boule blanche, on la remplace par une boule noire.
Une suite de tirages est donc une suite de "blanc" ou "noir".

a) Déterminez les suites possibles à partir d'un sac contenant b boules blanches et n noires pour de « petites valeurs »de b et n.

b) Montrez que toute suite de tirages est finie et qu'elles ont toutes la même longueur.

c) Donnez un algorithme récursif qui calcule le nombre de tirages possibles. Justifiez sa terminaison.

4) [Algorithme d'Euclide] Décrivez un algorithme de paramètres deux entiers naturels a, b qui calcule le pgcd d de ces deux entiers. Puis modifiez-le pour obtenir en plus deux entiers relatifs u, v tels que $au + bv = d$.

5) [Chiffres romains] Décrivez un algorithme qui donne la valeur entière d'un nombre écrit en chiffres romains ($I = 1$, $V = 5$, $X = 10$, $L = 50$, $C = 100$, $D = 500$, $M = 1000$).

6) [Partitions d'un entier] On appelle p-partition d'un entier $n \in \mathbb{N}^*$ tout p-uplet $(a_1, \ldots, a_p)$ d'entiers non nuls, tels que $a_1 \leqslant \ldots \leqslant a_p$ et $n = a_1 + \ldots + a_p$. Par exemple, $(1, 1, 2, 3, 3, 3)$ est une partition de 13.
Pour tout $p \in \mathbb{N}^*$, on pose $P(n, p)$ le nombre de p-partitions de n et $\Pi(n)$ le nombre de partitions de longueur quelconque de n, autrement dit $\Pi(n) = \sum_{p=1}^{\cdots} P(n, p)$.

a) Que valent $P(n, 1)$, $P(n, n)$ et $P(n, p)$ si $p > n$?

b) Pour $p \geqslant 2$, justifiez qu'il existe une bijection entre l'ensemble des $(p-1)$-partitions de $n-1$ et celui des p-partitions de n dont le premier terme est 1.

c) Justifiez qu'il existe une bijection entre l'ensemble des p-partitions de $n - p$ et celui des p-partitions de n dont le premier terme est au moins 2.

d) Déterminez une définition récursive de $P(n, p)$.

e) Décrivez un algorithme qui calcule $\Pi(n)$.

7) [Décomposition en facteurs premiers]

a) Décrivez un algorithme récursif de paramètre un entier naturel non nul n et qui calcule la partie entière de $\log_2 n$, c'est-à-dire le plus grand entier k tel que $2^k \leqslant n$.

b) Décrivez un algorithme récursif de paramètre un entier naturel n au moins égal à 2 et qui calcule le plus petit diviseur de n supérieur ou égal à 2.

c) Décrivez un algorithme qui calcule la décomposition en facteurs premiers d'un entier naturel $n \geqslant 2$: on se contentera d'afficher les facteurs premiers à l'écran.

8) [Opérations sur les tableaux]

a) Décrivez un algorithme de paramètre un tableau de nombres t et qui retourne la plus grande valeur de ce tableau.

b) Décrivez un algorithme de paramètres un tableau t et un objet x, qui retourne le nombre d'occurrences de x dans le tableau t.

c) Décrivez un algorithme de paramètres un tableau t et un objet x, qui retourne la première position de x dans t si elle existe, -1 sinon.

9) [Algorithme de Horner] Étant donné un polynôme p à coefficients « réels »représenté par un tableau de flottants et un nombre x, écrivez un algorithme de paramètres p et x qui calcule $p(x)$ selon l'idée suivante :
— si $p(x) = a_0 + a_1x + a_2x^2$, alors on calcule $p(x) = a_0 + x(a_1 + x(a_2))$
— si $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, alors on calcule $p(x) = a_0 + x(a_1 + x(a_2 + x(a_3)))$
En considérant les opérations algébriques sur les nombres comme élémentaires, quelle est la complexité de cet algorithme en fonction de $n = \deg p$?

10) [Division euclidienne dans $K[X]$] Avec la même représentation des polynômes (voir ci-dessus), décrivez un algorithme de paramètres deux tableaux représentent deux polynômes A et B et qui calcule le quotient et le reste de la division euclidienne de A par B. On supposera connues des fonctions permettant de faire des sommes, des produits de polynômes par des constantes ou des produits entre eux.