TP 10 du 25 mars 2020 (polynômes)

A tout polynôme $A = a_0 + a_1X + \cdots + a_nX^n$ on associe la liste $[a_0, a_1, \ldots, a_n]$ de ses coefficients. Dans la suite un polynôme sera représenté par une liste $[a_0, a_1, \ldots, a_n]$ de ses coefficients. Le polynôme $5 + 3X + 4X^2$ pourra être représenté au choix par les listes $[5, 3, 4]$, $[5, 3, 4, 0]$, $[5, 3, 4, 0, 0]$ $\ldots$, la liste $[5, 3, 4]$ est appelé forme normale de ce polynôme.

I Généralités

Q1: Ecrire une fonction **normale(A)** qui modifie la liste A en supprimant les éventuels zéros en fin de liste.et renvoie cette liste. Par exemple **normale([5,3,4,0,0,0])** doit renvoyer la liste [5,3,4] et **normale([0,0,0])** doit renvoyer la liste vide.

Q2: Ecrire une fonction **degree(A)** qui, si A est non nul, renvoie le degré de A et qui renvoie -1 sinon Par exemple **degree([5,3,4,0,0,0])** doit renvoyer 2 et **degree([0,0,0])** doit renvoyer $\mathbf{-1}$..

Q3a: Ecrire une fonction **evaluation(A,a)** qui renvoie la valeur de A(a) Par exemple **evaluation([5,3,4,0,0,0],2)** doit renvoyer $5 + 3 \times 2 + 4 \times 2^2$ soit 27.

Q3b: On voudrait éviter le calcul de puissances dans la fonction précédente car une puissance est plus longue à l'exécution qu'une simple multiplication. Réécrire une fonction **evaluation(A,a)** qui renvoie la valeur de A(a) sans utiliser de puissance et avec $O(n)$ multiplications.

Q4: Ecrire une fonction **somme(A,B)** qui renvoie la forme normale du polynôme A+B (si les deux listes A et B sont de longueur differentes, on commencera par ajouter des zéros en fin de liste du polynôme de plus bas degré de manière que les deux listes soient de longueur égales.
Par exemple **somme([5,3,4],[1,2,3,4])** doit renvoyer **[6,5,7,4]** et **somme([5,3,4,-4],[1,2,3,4])** doit renvoyer **[6,5,7]**.

II Produit de polynômes

Si $A = a_0 + a_1X + \cdots + a_nX^n$ alors le polynôme $A \times X^k$ est égal à $a_0X^k + a_1X^{k+1} + \cdots + a_nX^{k+n}$.

Q5: Ecrire une fonction **produit1(A,k,c)** qui renvoie la forme normale du polynôme $A \times cX^k$. Par exemple **produit1([5,3,4],2,1)** doit renvoyer **[0,0,5,3,4]** et **produit1([5,3,4],2,-2)** doit renvoyer **[0,0,-10,-6,-8].**

On souhaite se servir de la fonction précédente pour programmer le produit de deux polynômes quelconques. Pour cela, on remarque que si $B = b_0 + b_1X + \cdots + b_pX^p$ alors, par distributivité,

$$A \times B = (b_0 \times A) + (b_1 \times A \times X) + \cdots + (b_p \times A \times X^p) .$$

Q6: Ecrire une fonction **produit(A,B)** qui renvoie la forme normale du polynôme $A \times B$ en utilisant la fonction **produit1**. Par exemple **produit1([2,3,4,0,5],[3,0,1])** doit renvoyer **[6,9,14,3,19,0,5]** car
$\left(2 + 3X + 4X^2 + 5X^4\right)\left(3 + X^2\right) = 6 + 9X + 14X^2 + 3X^3 + 19X^4 + 5X^6$.
Quelle est la complexité de cette fonction?

Q7a: Question de maths: Justifier qu'il existe un un seul polynôme de degré 6 vérifiant $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 0$ et $P(7) = 1$. Question informatique: déterminer ses coefficients.

Q7b: On voudrait utiliser ce qui précède pour obtenir la liste des coefficients binomiaux: $L = \left[\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}\right]$.
Pour cela on remarque que $(1 + X)^n = \binom{n}{0}1^nX^0 + \binom{n}{1}1^{n-1}X + \cdots + \binom{n}{n}1^0X^n = \binom{n}{0} + \binom{n}{1}X + \cdots + \binom{n}{n}X^n$ et donc la liste L est la liste des coefficients binomiaux L souahaitée.
Ecrire une fonction **liste(n)** qui renvoie cette liste.
Vérifier expérimentalement que pour tout $n \leq 100$ le plus grand coefficient binomial de la liste L se trouve toujours au milieu de cette liste.
(on peut démontrer cette propriété à l'aide d'une relation que vous avez peut-être vue en cours).

III Division euclidienne

Dans cette partie, on reprend les idées de la démonstration de l'existence de la division euclidienne des polynômes qui est une démonstration algorithmique.

Si $A = a_0 + a_1X + \cdots + a_nX^n$ et $B = b_0 + b_1X + \cdots + b_pX^p$ (avec $n \geq p$ et $b_p \neq 0$) alors la première étape de la division euclidienne consiste à calculer $A_1 = A - cX^kB$ de manière que $\deg(A_1) < \deg(A)$.

Q8: Comment faut-il choisir λ et k? Ecrire une fonction **suivant(A,B)** qui renvoie la liste [c,k,A1].

Q9: Ecrire une fonction **division(A,B)** qui renvoie la liste [Q,R] où Q est le quotient et R est le reste dans la division euclidienne de A par B. Par exemple **division([6,9,14,3,19,0,5],[3,0,2])** doit renvoyer **[[2,3,4,0,5],[]]** et **division([7,10,14,3,19,0,5],[3,0,2])** doit renvoyer **[[2,3,4,0,5],[1,2]].**
Indication: On commencera par traiter le cas où le degré de A est strictement inférieur au degré de B. Dans le cas contraire, on commencera par définir une liste Q dont la taille correspondant au quotient voulue et initialisée avec des valeurs nulles.

IV Produit de nombres entiers en binaire

Lorsque le processeur d'un ordinateur effectue le produit de nombres entiers positifs, il s'inspire de l'algorithme de multiplication des polynômes vu en section 2. On sait que les nombres sont codés en machine avec leur expression en base 2 (binaire). Le produit de deux nombres $n = a_0 + a_12^1 + \cdots + a_n2^n$ et $b = b_0 + b_12 + \cdots + b_p2^p$ peut donc se faire en appliquant la distributivité: $n \times m = b_0 \times n + b_1 \times 2 \times n + \cdots + b_p \times 2^p \times n$. Comme les chiffres binaires b_i valent 0 ou 1, la somme précédente est une somme de termes de la forme $2^i \times n$. Or si $n = a_0 + a_12^1 + \cdots + a_n2^n$ alors $2^i \times n = a_02^i + a_12^{i+1} + \cdots + a_n2^{i+n}$ qui est l'écriture de 2^in en base 2 (les a_i étant supposés au départ dans $\{0, 1\}$). Il suffit finalement de sommer les différents termes de la forme $2^i \times n$ pour obtenir la valeur de $n \times m$.

Dans la suite, les entiers n et m sont supposés donnés par les listes $[a_0, a_1, \ldots, a_n]$ et $[b_0, b_1, \ldots, b_p]$ de leurs chiffres binaires.

Les listes sont données dans l'ordre croissant des puissances de 2 pour simplifier (et ressembler à ce qui a été fait pour les polynômes). Cela correspond à l'écriture binaire dans l'ordre inverse.

Q10: Expliquer pourquoi utiliser simplement la fonction **produit(A,B)** déja écrite pour faire le produit de deux entiers m et n ne donne pas toujours un résultat correct. A quel condition **produit(n,m)** donne-t-elle un résultat correct?

Q11: En adaptant ce qui a été fait dans la section 2, écrire une fonction **produit(n,m)** qui renvoie le produit $n \times m$ sous forme binaire. (On pourra s'aider de la fonction python **bin** pour vérifier ses résultats).