

Problème 1 : produit de polynômes

On représentera un polynôme par la liste de ses coefficients : le polynôme $P = a_0 + a_1X + \dots + a_nX^n$ sera représenté par la liste $\mathbf{p} = [a_0, a_1, \dots, a_n]$. Un polynôme de degré d pourra donc être représenté par une liste de longueur $d + 1$. Une telle représentation n'est pas unique : $2X^2 + 1$ peut être représenté par les listes $[1, 0, 2]$ ou $[1, 0, 2, 0, 0]$ par exemple. On représentera le polynôme nul par la liste $[0]$ (ainsi les listes ne seront jamais vides) et on prendra comme convention que le degré du polynôme nul est -1 .

1. On considère la fonction suivante :

```
def deg(p) :
    # à compléter
    else :
        a = p[-1]
        if a != 0 :
            return len(p)-1
        else :
            p.pop()
            return deg(p)
```

- Compléter la fonction précédente de façon à ce qu'elle renvoie le degré du polynôme représenté par la liste \mathbf{p} , supposée non vide (rajouter autant de lignes que nécessaire).
- On suppose que $\mathbf{p} = [0, 1, 0, 2, 0]$. Que contient la liste \mathbf{p} après l'exécution de `deg(p)` ?

Dans la suite de cet exercice, on s'intéresse au produit de deux polynômes P et Q . Quitte à rajouter des coefficients nuls, on supposera que P et Q sont représentés par deux listes de même longueur $n + 1$ et que $n = 2^k$ est une puissance de 2. Les fonctions demandées ne devront pas nécessairement supprimer les zéros inutiles dans les listes représentant les polynômes.

Pour évaluer le coût des fonctions, on ne comptera que les multiplications faites sur les coefficients des polynômes.

- Écrire une fonction `sum(p,q)` qui renvoie la liste représentant le polynôme somme de P et Q . On rappelle que P et Q sont supposés représentés par deux listes de même longueur.
- Écrire une fonction `prodScal(p,a)`, qui prend en argument une liste \mathbf{p} représentant un polynôme P et un flottant a (associé à un scalaire a), et qui renvoie la liste représentant le polynôme aP .
- Écrire une fonction `prodMonome(p,r)`, qui prend en argument une liste \mathbf{p} représentant un polynôme P et un entier naturel r , et qui renvoie la liste représentant le polynôme $X^r P$.

3. Si $P = \sum_{k=0}^n p_k X^k$ et $Q \in \mathbb{R}_n[X]$, on peut effectuer le produit de P par Q en remarquant que $PQ = \sum_{k=0}^n (p_k X^k \times Q)$

- Si P et Q sont représentés par des listes de longueur $n + 1$, quelle est, à priori, la longueur de la liste représentant le produit PQ ?
 - Écrire une fonction `mult(p,q)` qui prend en argument deux listes représentant des polynômes et qui renvoie la liste représentant leur produit ; cette fonction devra utiliser les fonctions `sum`, `prodScal` et `prodMonome`
 - Combien de multiplications sont réalisées par cette fonction en fonction de n , le « degré » des polynômes ? (on rappelle que l'on a en fait $\deg(P) \leq n$)
4. On cherche à écrire une fonction réalisant le produit de deux polynômes plus efficace que la précédente. On n'utilisera donc pas la fonction `mult` dans la suite.
- Si P est de degré $\leq n = 2^k$, on note Q et R le quotient et le reste de la division euclidienne de P par $X^{n/2}$:

$$P = X^{n/2}Q + R \quad \text{avec} \quad \deg(R) < \frac{n}{2}$$

Écrire une fonction `decompose(p)`, qui prend en argument une liste représentant P (donc de longueur $n + 1$), et qui renvoie les listes représentant Q et R définis au dessus ; Q et R doivent être représentés, tous les deux, par des listes de longueur $1 + n/2$.

- Si $P_1 = X^{n/2}Q_1 + R_1$ et $P_2 = X^{n/2}Q_2 + R_2$ sont les divisions euclidiennes de P_1 et P_2 (de degrés $\leq n$), vérifier que

$$P_1 P_2 = X^n Q_1 Q_2 + X^{n/2}((Q_1 + R_1)(Q_2 + R_2) - Q_1 Q_2 - R_1 R_2) + R_1 R_2$$

- À l'aide de l'égalité précédente, écrire une fonction réursive `prod(p,q)` qui renvoie la liste représentant le produit de deux polynômes P et Q ; cette fonction devra obligatoirement utiliser les fonctions `decompose`, `prodMonome`, `prodScal` et `sum`.
Évaluer le coût de cette fonction.

Problème 2 : Algorithme de Kaprekar

1. On considère la fonction $f(n, p)$ suivante ; n et p sont deux entiers naturels tels que $n < 10^p$.

```
def f(n, p) :
    L = []
    for k in range(p) :
        n, r = n//10, n%10
        L.append(r)
    return L
```

- a) Exécuter $f(123, 3)$ et $f(22, 3)$; remplissez un tableau faisant apparaître les valeurs de k, n, r, L au cours de l'exécution et la valeur renvoyée.
- b) Quel est le rôle de cette fonction ?
2. Écrire une fonction $occ(L)$ qui prend en argument une liste L d'entiers de $\llbracket 0, 9 \rrbracket$ (donc des chiffres), supposée non vide, et qui renvoie une liste F , de longueur 10, telle que $F[k]$ est le nombre de fois où le chiffre k apparaît dans la liste L .
- Par exemple $occ([0, 1, 1, 2, 5, 1, 7, 3, 0, 4, 5, 0])$ devra renvoyer $[3, 3, 1, 1, 1, 2, 0, 1, 0, 0]$.
3. On se donne deux entiers n et p tels que $n < 10^p$. On suppose posséder une fonction $chiffres(n, p)$ qui renvoie la liste des p chiffres de l'entier n (en rajoutant des zéros si besoin) ; $chiffres(121, 4)$ renvoie $[1, 2, 1, 0]$ (l'ordre des chiffres n'a pas d'importance). À partir des chiffres de l'entier n , on écrit les deux nombres M et m tels que M (*resp.* m) soit le plus grand (*resp.* le plus petit) des nombres que l'on peut écrire avec les p chiffres de n .
- Par exemple si $n=6106$ et $p=5$, la liste des p chiffres de n est $[6, 1, 0, 6, 0]$; on a alors $M=66100$ et $m=166$ (ou $m=00166$). On peut remarquer que pour écrire les nombres M et m , il suffit de ranger les p chiffres de n dans l'ordre croissant ou décroissant.
- a) Si on suppose que $occ(chiffres(n, p))$ renvoie la liste $[2, 0, 1, 0, 2, 1, 0, 3, 1, 0]$, quelles sont les valeurs des entiers M et m associés à n ?
- b) Écrire une fonction $K(n, p)$ qui renvoie le couple (M, m) associé aux p chiffres de n .
4. L'algorithme de Kaprekar consiste, à partir d'un entier n , écrit sur p chiffres, à calculer les entiers M et m précédents, puis à recommencer à partir de l'entier $M-m$ jusqu'à retrouver une valeur déjà rencontrée (une fois cette répétition atteinte, les valeurs suivantes décrivent un cycle). À partir de $n=42$, avec $p=2$, les valeurs obtenues sont les suivantes

n	M	m	$M-m$
42	42	24	18
18	81	18	63
63	63	36	27
27	72	27	45
45	54	45	9
9	90	9	81
81	81	18	63

Écrire une fonction $Kaprekar(n, p)$ qui prend en argument deux entiers n et p (tels que $n < 10^p$) et qui renvoie la liste des différentes valeurs de $M-m$ définies précédemment, jusqu'à l'apparition d'une répétition. On rappelle que *elt in L* renvoie *True* ou *False* selon que *elt* appartient ou non à la liste L .

$Kaprekar(42, 2)$ devra renvoyer $[42, 18, 63, 27, 45, 9, 81, 63]$