

Mathématiques Discrètes

feuille numéro 08

9 novembre 2019

Exercice 1

dans tout cet exercice n désigne un entier naturel.

Q 1.1 Compter le nombre de mots avec exactement n parenthèses ouvrantes et n parenthèses fermantes.

Q 1.2

Vérifier que par un calcul que

$$\frac{\binom{2n}{n}}{n+1} = \binom{2n}{n} - \binom{2n}{n+1}$$

Q 1.3 Compter le nombre de mots à $n+1$ parenthèses fermantes et $n-1$ parenthèses ouvrantes. On appelle \mathcal{M}_n l'ensemble de ces mots.

On donne le code python suivant :

```
def premier_defaut(m):
    """
    détermine l'indice du premier défaut de bon parenthésage
    :param m: la chaîne à tester
    :type m: str
    :return: l'indice du premier défaut
    :rtype: int
    CU. m est une chaîne ne contenant que des parenthèses

    >>> premier_defaut('')
    0
    >>> premier_defaut('()()()')
    4
    >>> premier_defaut('()()()')
    4
    """
    assert type(m) == str , "m doit être une chaîne"
    assert set(m).issubset({ '(', ')' }) , "m ne doit contenir que des parenthèses"
    compteur = 0
    i = 0
    while i<len(m) and compteur>=0:
        if m[i]=='(':
            compteur += 1
        else:
            compteur -= 1
        i += 1
    assert compteur != 0 , "m est bien parenthésé"
    return i-1
```

Q 1.4 Expliquez pourquoi un appel à `premier_defaut` avec n'importe quel mot de \mathcal{M}_n ne peut pas déclencher `AssertionError`

On donne le code suivant :

```
ASSOCIATION= { '(' : ')', ')' : '(' }
def complement(m):
    """ ... doc omise volontairement """
    return "".join([ASSOCIATION[c] for c in m])

def transforme(m):
    """ ... doc omise volontairement """
    i=premier_defaut(m)
    return m[:i+1]+complement(m[i+1:])
```

Q 1.5 Si m est un mot bien parenthésé que dire de l'expression `transforme(m)` ?

Q 1.6 Si m est un mot mal parenthésé que dire de l'expression `transforme(m)`. Précisez notamment la longueur ce mot. Peut on prévoir la valeur de `premier_defaut(transforme(m))` ?

Q 1.7 Si m est un mot mal parenthésé que dire de l'expression `transforme(transforme(m))` ?

Q 1.8 Si m est un mot de \mathcal{M}_n que dire de l'expression `transforme(m)` ? Précisez notamment le nombre de parenthèses ouvrantes et le nombre de parenthèses fermantes.

Q 1.9 Soit m un mot mal parenthésé possédant n parenthèses ouvrantes et n parenthèses fermantes, que dire de `transforme(m)`

Q 1.10 Prouver qu'il y a autant de mots possédant n parenthèses ouvrantes et n parenthèses fermantes et mal parenthésés, que de mots de \mathcal{M}_n .

Q 1.11 En déduire le nombre de mots possédant n parenthèses ouvrantes et n parenthèses fermantes et bien parenthésés.

Exercice 2

Le langage de programmation BRAINFUCK est un langage de programmation sur un alphabet à 8 symboles.

Q 2.1 n désigne un entier naturel. Combien y a-t-il de mots de n lettres sur cet alphabet ?

L'alphabet considéré contient un crochet ouvrant et un crochet fermant. La seule règle de syntaxe est que les crochets ouvrants "[" et crochets fermants "]" doivent se correspondre, formant ainsi un mot bien parenthésé.

Q 2.2 Combien de mots bien parenthésés, différents et de longueur $2k$ contenant uniquement des crochets peut-on écrire ?

On suppose $k \leq \frac{n}{2}$.

Q 2.3 Combien de mots de longueur $n - 2k$ peut-on écrire avec les 6 symboles restants ?

Q 2.4 Combien de programmes BRAINFUCK de longueur n , possédant k crochets ouvrants, syntaxiquement corrects, peut-on écrire ?

Q 2.5 Combien de programmes BRAINFUCK de longueur n , syntaxiquement corrects, peut-on écrire ? On donnera le résultat sous la forme d'une somme qu'on ne cherchera pas à calculer.

Exercice 3

n désigne un entier naturel.

Q 3.1

Combien y a-t-il de chemins de $(0,0)$ à (n,n) en ne faisant que des pas du type $(i,j) \mapsto (i+1,j)$ ou bien $(i,j) \mapsto (i,j+1)$ et en restant sous la diagonale?

Exercice 4

Q 4.1 Combien y a-t-il d'arbres binaires **complets** à cinq nœuds internes et six nœuds externes.

Q 4.2 Combien y a-t-il de manière de décorer les cinq nœuds internes d'un arbre binaire complet avec les 4 opérateurs choisis dans l'ensemble $E = \{+, -, *, /\}$?

Q 4.3 Combien y a-t-il de manière de choisir un arbre binaire complet à 5 nœuds internes et six nœuds externes et dont les cinq nœuds internes sont décorés avec les 4 opérateurs choisis dans l'ensemble $E = \{+, -, *, /\}$?

Le « *compte est bon* » est un jeu où on doit atteindre un nombre donné à partir de 6 nombres donnés et en s'autorisant d'utiliser les quatre opérations arithmétiques usuelles (addition, soustraction, multiplication et division). Pour les divisions, il faut que le dividende soit divisible par le diviseur. Par exemple avec 4, 7, 3, 5, 11 et 2 on peut réaliser 2015 ainsi

$$(((4 * 7) + 3) * 5) * (11 + 2)$$

Q 4.4 En remarquant que les calculs peuvent être mis sous la forme d'un arbre binaire dont les nœuds internes sont les opérations arithmétiques, et les nœuds externes les nombres donnés, donnez une majoration du nombre de cas à envisager lors de la résolution par une recherche exhaustive.

Exercice 5

On désigne par n un entier.

Le but de cet exercice est de compter de plusieurs manières le nombre B_n de mots sur l'alphabet $X = \{ ' (' , ' [' , '] ' , ') ' \}$ bien parenthésés et de longueur $2n$.

Version 1

Q 5.1 Combien y a-t-il de mots bien parenthésés de longueur $2n$ constitué uniquement des parenthèses $' (' , ') '$?

Q 5.2 On note k un entier compris au sens large entre 0 et n .

Combien y a-t-il de mots bien parenthésés avec k crochets ouvrants $' ['$ et $n - k$ parenthèses ouvrantes $' ('$. Indication : On pourra s'aider de la question précédente, et se dire qu'il suffit de choisir k parenthèses ouvrantes $' ('$ pour les transformer en crochet $' ['$... Dans la suite, on note B_n^k ce nombre.

Dans la suite, on note B_n^k ce nombre.

Q 5.3 En déduire une expression de B_n sous la forme d'une somme $(\Sigma \dots)$. On n'essiera pas de calculer cette somme.

Version 2

On considère l'application ϕ définie de l'ensemble des mots bien parenthésés sur X de longueur $2n$ dans l'ensemble des mots bien parenthésés sur l'alphabet $\{ ' (' , ') ' \}$ obtenue en remplaçant crochet ouvrant $' ['$ par parenthèse ouvrante $' ('$ et crochet fermant $'] '$ par parenthèse fermante $') '$

Q 5.4 Montrez que cette application est surjective.

Q 5.5 Quelle est la nature de la relation \mathcal{R} définie sur l'ensemble des mots bien parenthésés sur X par $m_1 \mathcal{R} m_2$ ssi $\phi(m_1) = \phi(m_2)$.

Q 5.6 soit m un mot de l'ensemble Y_n (les mots bien parenthésés sur $\{ ' (' , ') ' \}$ et de longueur $2n$). Quel est le cardinal de $\{x \in X_n | \phi(x) = m\}$

Q 5.7 En déduire une autre expression du nombre B_n

Version 3

Q 5.8 Établir l'équation de récurrence

$$B_{n+1} = \sum_{k=0}^{k=n} 2B_k B_{n-k}$$

Q 5.9 Effectuer le changement de variable

$$B_k = 2^k X_k$$

En déduire l'équation de récurrence vérifiée par X_n .

Q 5.10 Retrouver l'expression de B_n .