L3 maths

CT : Coefficients multinomiaux

Durée de l'épreuve : 2h00

L'utilisation de toute calculatrice, téléphone, ordinateur ou dispositif électronique est interdite pendant l'épreuve

Le sujet est volontairement long et il n'est pas nécessaire de le traiter entièrement pour pouvoir obtenir la note 20/20.

Lorsqu'un programme en python est demandé, l'important n'est pas de savoir s'il s'exécute correctement sans erreur de syntaxe, mais si les opérations effectuées répondent à la question.

Votre attention est attirée sur la rigueur attendue lorsqu'une démonstration mathématique est demandée.

Étant donné un ensemble E, on dit que les sous-ensembles P_1, P_2, ..., P_m "forment une partition de E" si E est leur union disjointe, c'est à dire si $\forall j \in E, \exists! k \in [\![1;m]\!] : j \in P_k$ (et $\forall k \in [\![1;m]\!] : P_k \subset E$).

On rappelle que si `L` est une liste alors `L[0]` désigne le premier élément de la liste, `L[1]` désigne le deuxième élément de la liste, etc. Et `L[-1]` désigne le dernier élément.

On rappelle aussi que `L[:i]` désigne la sous liste `[L[0],L[1],...,L[i-1]]`, et que `L[i:]` désigne la sous liste `[L[i],L[i+1],...,L[-1]]`.

On rappelle enfin que `range(n)` désigne $[\![0\,;\,n[\![$ si $n > 0$, et l'ensemble vide sinon.

I. Introduction

1. Définition

Soit $m \in \mathbb{N}^*$. On considère les trois fonctions A, B, et C de $\mathbb{N}^m$ dans $\mathbb{N}$ définies ci-dessous (on montrera par la suite que ces trois fonctions sont égales).

1.1) Coefficient de polynôme

Étant donnés $(k_1,k_2,\ldots,k_m) \in \mathbb{N}^m$, on pose $n = \sum_{i=1}^{m} k_i$, et on considère l'expression $(x_1 + x_2 + x_3 + \cdots + x_m)^n$. Cette expression est un polynôme (en m variables x_1, ..., x_m), et on définit $A(k_1,k_2,\ldots,k_m)$ comme étant égal au coefficient de $x_1^{k_1}x_2^{k_2}x_3^{k_3}\ldots x_m^{k_m}$ dans ce polynôme.

1. Développer le polynôme $(x_1 + x_2 + x_3)^2$. En déduire les valeurs de $A(2,0,0)$, $A(0,2,0)$, $A(0,0,2)$, $A(1,1,0)$, $A(1,0,1)$ et $A(0,1,1)$.

1.2) Nombre de partitions

Étant donnés $(k_1,k_2,\ldots,k_m) \in \mathbb{N}^m$, on pose $n = \sum_{i=1}^{m} k_i$, et on considère l'ensemble $E = [\![1;n]\!]$. On s'intéresse aux partitions de E en m sous-ensembles P_1, P_2, ... P_m.

$B(k_1,k_2,\ldots,k_m)$ est le nombre de partitions telles que $\mathrm{Card}(P_1) = k_1$, $\mathrm{Card}(P_2) = k_2$, etc.

2. Démontrer qu'en tout, il y a m^n partitions de l'ensemble $E = [\![1;n]\!]$ en m sous ensembles P_1, P_2, ... P_m. *Indication : on pourra exhiber une bijection avec l'ensemble* $[\![1;m]\!]^n$.
3. Dans le cas où $n = 2$ et $m = 3$, lister toutes ces partitions et en déduire les valeurs de $B(2,0,0)$, $B(0,2,0)$, $B(0,0,2)$, $B(1,1,0)$, $B(1,0,1)$ et $B(0,1,1)$.

1.3) Expression en terme de factorielles

Enfin, on définit la fonction C par

$$C(k_1,k_2,\ldots,k_m) = \frac{(k_1+k_2+\cdots+k_m)!}{(k_1!)\times(k_2!)\times\cdots\times(k_m!)}$$

4. Calculer $C(2,3,2)$. *Indication : pour simplifier le calcul (et l'effectuer sans calculatrice), on pourra chercher à écrire* $C(2,3,2)$ *comme un produit de coefficients binomiaux.*
5. Démontrer que $\forall (k_1,k_2,\ldots,k_m) \in \mathbb{N}^m, \quad C(k_1,k_2,\ldots,k_m) \in \mathbb{N}$.

2. Relations de récurrence

Cette partie va montrer que les trois fonctions A, B, et C définies ci-avant sont égales, et identifie deux relations de récurrences qu'elles satisfont. *Si vous ne parvenez pas à montrer certains résultats de cette partie, vous pouvez les admettre afin de traiter la suite du sujet.*

6. Montrer que les fonctions A, B, et C définies ci-avant sont égales entre elles et satisfont toutes trois la relation de récurrence ci-dessous :

$$\forall m \geqslant 2, \quad \forall (k_1,\ldots k_m) \in \mathbb{N}^m, \quad F(k_1,k_2,\ldots,k_m) = \binom{n}{k_1} F(k_2,\ldots,k_m)$$

où $n = \sum_{i=1}^{m} k_i$ (et où F désigne chacune des fonctions A, B, et C).
Vous pourrez montrer d'abord la relation de récurrence puis en déduire qu'elles sont égales, ou bien l'inverse, selon ce qui vous semble le plus simple/le plus pertinent.

7. Montrer que de plus, elles satisfont la relation :
$\forall (k_1,\ldots k_m) \in (\mathbb{N}^*)^m$,
$F(k_1,k_2,\ldots,k_m) = F(k_1-1,k_2,k_3,\ldots,k_{m-1},k_m) + F(k_1,k_2-1,k_3\ldots,k_{m-1},k_m)$
$+\cdots+F(k_1,k_2,\ldots,k_{m-1},k_m-1)$
8. Enfin, montrer que "si un des k_i est nul, on peut l'enlever", c'est à dire qu'elles satisfont $\forall (k_1,\ldots k_m) \in \mathbb{N}^m, \quad \forall i \in [\![0;m]\!] : \quad F(k_1,k_2,\ldots,k_i,0,k_{i+1},k_{i+2},\ldots,k_m) = F(k_1,k_2,\ldots,k_m)$.

3. Notation

Les valeurs de cette fonction A (ou B ou C, cela revient au même) s'appellent "coefficients multinomiaux".
On pose (comme précédemment) $n = \sum_{i=1}^{m} k_i$ et on note $\binom{n}{k_1,k_2,\ldots,k_m} = A(k_1, k_2, \ldots, k_m)$.

II. Calcul des coefficients multinomiaux en python

On définit ci-dessous deux fonctions `multinomial` en Python telles que pour `k=[`k_1`, `k_2`, ..., `k_m`]`, `multinomial(k)` est sensé calculer $\binom{n}{k_1,k_2,\ldots,k_m}$ (où $n = \sum_i k_i$).

9. Pour chacune de ces fonctions (proposées ci-dessous) déterminer ce que calcule python et ce qu'il renvoie quand on exécute `multinomial([1,1,0])`.
   Argumenter ensuite pour déterminer si la fonction proposée est correcte. Si elle n'est pas correcte, proposez une modification qui la rend correcte.

1. Première version

```
def multinomial(k):
    ## calcul de la liste l=[0!,1!,2!,...,n!]
    l=[1]
    for k_i in k:
        for i in range(k_i):
            l.append(l[-1]*len(l))
    ## calcul du denominateur k1!*k2!*...*km!
    denominateur=1
    for k_i in k:     denominateur=denominateur*l[k_i]
    ## calcul du numérateur n!
    numerateur=l[len(l)]
    ## resultat:
    return numerateur/denominateur
```

2. Deuxième version

```
def multinomial(k):
    ## Si un des ki est nul, l'enlever
    for i in range(len(k)):
        if k[i]==0:      return multinomial(k[:i]+k[i+1:])
    ## sinon utiliser la relation de récurrence
    res=0
    for j in range(len(k)):
        res=res+multinomial([(k[i] if i!=j else k[i]-1) for i in range(len(k))])
    return res
```

III. Loi multinomiale

On considère la situation où on lance n dés (chaque dé a 6 faces, numérotées de 1 à 6). On désigne par X_1 le nombre de dés qui tombent sur la face 1, par X_2 le nombre de dés qui tombent sur la face 2, etc.

On peut montrer qu'on a alors

$$\mathbb{P}[X_1 = k_1, X_2 = k_2, \ldots, X_6 = k_6] = \binom{n}{k_1, k_2, \ldots, k_6} \Big/ 6^n$$

On cherche à vérifier cette affirmation en effectuant des simulations. On définit tout d'abord la fonction `de()` qui simule un dé et renvoie un nombre au hasard entre 1 et 6 :

```
def de():
    import random
    return random.choice(range(1,7))
```

10. Écrire une fonction `des(n)` qui lance `n` fois un dé et renvoie les valeurs de $(X_1, X_2, \ldots, X_6)$, c'est à dire le nombre de dés qui ont fait 1, le nombre qui ont fait 2, etc.
11. Écrire une fonction `mult(m,n)` qui détermine tous les m-uplets $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$ tels que $\sum_{i=1}^{m} k_i = n$. *Si on préfère, on pourra les considérer en python comme des* `list` *(listes) au lieu de* `tuple` *(multiplets).*
12. Écrire une fonction `stats_de(n,N)` qui répète `N` fois la fonction `des(n)` (le lancer de n dés) et qui pour chaque 6-uplet $(k_1, k_2, \ldots, k_6) \in \mathbb{N}^6$ (tel que $\sum_{i=1}^{6} k_i = n$) indique quelle proportion de fois (parmi les `N`) on a eu $(X_1, X_2, \ldots, X_6) = (k_1, k_2, \ldots, k_6)$.

    Exécuter cette fonction sur ordinateur pour de très grandes valeurs de N permettrait de se convaincre que $\mathbb{P}[X_1 = k_1, X_2 = k_2, \ldots, X_6 = k_6]$ *est bien égal à* $\binom{n}{k_1, k_2, \ldots, k_6}/6^n$.