Chapitre 3 : Résumé sur le dénombrement.

I Cardinal d'un ensemble fini

Définition 1 (Cardinal)

Le cardinal d'un ensemble fini est simplement le nombre d'éléments de cet ensemble.
Par convention l'ensemble vide est fini et de cardinal 0.

Proposition 1

Soit E un ensemble fini et soit $A \subset E$.

- Soit $\bar{A}$ le complémentaire de A dans E. Alors $\bar{A}$ est fini et

$$\text{Card}(\bar{A}) = \text{Card}(E) - \text{Card}(A)$$

- Soit $n \in \mathbb{N}^*$ et $(A_1, \cdots A_n)$ une famille d'ensembles finis deux à deux disjoints, on a :

$$\text{Card}\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} \text{Card}\,(A_i)$$

Définition 2 (Produit cartésien : principe multiplicatif.)

Soient k un entier supérieur à 2 et $E_1, E_2, ..., E_k$, k ensembles **non vides**.
L'ensemble des **k-listes** (ou **k-uplets**) **ordonnées** de la forme $(x_1, x_2, \cdot, x_k)$, avec $x_i \in E_i$ pour tout $i \in [\![1, k]\!]$ est appelé le **produit cartésien** des ensembles $E_1, E_2, ..., E_k$ et est noté :

$$E_1 \times \cdots \times E_k = \{(x_1, x_2, \cdots, x_k) \mid x_1 \in E_1, x_2 \in E_2, \cdots, x_k \in E_k\}$$

Les 2-listes sont appelées des couples et les 3-listes des triplets.

Proposition 2 (Produit cartésien : principe multiplicatif.)

Soit E et F deux ensembles non-vides. Le produit cartésien $E \times F$ est fini si et seulement si A et B sont tous les deux finis et alors

$$\text{Card}(E \times F) = \text{Card}(E) \times \text{Card}(E)$$

Plus généralement soit $n \in \mathbb{N}^*$ et $(E_1, \cdots, E_n)$ une famille d'ensembles non vides. Le produit cartésien $E_1 \times \cdots \times E_n$ est un ensemble fini si et seulement si tous les E_i sont finis et alors

$$\text{Card}(E_1 \times \cdots \times E_n) = \prod_{k=1}^{n} \text{Card}(E_k)$$

II p-listes d'un ensemble

Définition-Proposition 3 (p-liste avec répétition d'un ensemble)

Pour le cas particulier A^p avec $p \in \mathbb{N}^*$, les éléments de l'ensemble A^p sont appelés des **p-liste** (avec répétitions possibles) d'éléments de A et l'on a :

$$\text{Card}(A^p) = \text{Card}(A)^p$$

Définition 3 (p-listes sans répétition d'un ensemble.)

Soit F un ensemble et $p \in \mathbb{N}^*$. On appelle p-liste sans répétition de F toute p-liste $(x_1, \cdots, x_p)$ d'éléments de F deux-à-deux distincts.

Théorème 4 (Nombre de p-listes sans répétition d'un ensemble.)

Soit F un ensemble fini de cardinal $n \in \mathbb{N}$ et soit $p \in \mathbb{N}$. Alors

- Si $p \leqslant n$, il y a $\dfrac{n!}{(n-p)!}$ p-listes sans répétition d'éléments de F.
- Si $p > n$, il y a 0 p-listes sans répétition d'éléments de F.

Définition-Proposition 5 (Permutations)

Soit F un ensemble fini de cardinal $n \in \mathbb{N}$. Une n-liste est appelée une **permutation** des éléments de F.
Le nombre de permutation d'un ensemble à n éléments est : $n!$

III Combinaisons.

Définition 4

Soit E un ensemble fini et $k \in \mathbb{N}$. On appelle k-combinaison de E toute partie de E à k éléments

Théorème 6 (Nb de sous parties à k éléments : k-combinaisons)

Soit E un ensemble fini de cardinal $n \in \mathbb{N}$ et soit $k \in [\![0, n]\!]$. Le nombre de k-combinaisons de E est le nombre $\binom{n}{k}$ donné par

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Par convention, si $k < 0$ ou $k > n$ on pose $\dbinom{n}{k} = 0$. On le lit « k parmi n ».

On appelle les nombres $\dbinom{n}{k}$ des **coefficients binomiaux** (la raison viendra plus tard).

Proposition 7 (Propriétés des "coefficients binomiaux".)

Soient $n \in \mathbb{N}$ et $k \in [\![0; n]\!]$, on a les propriétés :

- $\dbinom{n}{0} = \dbinom{n}{n} = 1$
- $\dbinom{n}{k} = \dbinom{n}{n-k}$
- $\dbinom{n+1}{k+1} = \dbinom{n}{k} + \dbinom{n}{k+1}$ (ici k<n, cette formule est la formule de Pascal)
- $\dbinom{n}{k} = \dfrac{n-k+1}{k}\dbinom{n}{k-1}$ (Ici $k > 0$, $12^{ième}$ formule du traité de Pascal)

Proposition 8 (Nombre de sous-ensembles.)

Soit E un ensemble fini de cardinal $n \in \mathbb{N}$. L'ensemble $\mathcal{P}(E)$ est fini et on a

$$\mathrm{Card}(\mathcal{P}(E) = \sum_{k=0}^{n}) = 2^n$$

IV Démonstrations à connaitre.

Démonstration à connaitre absolument 1 (Formule du triangle de Pascal)

Si l'on commence à construire l'arbre représentant $n+1$ expériences de Bernoulli. Pour obtenir $k+1$ succès, nous pouvons décompter :

- Le nombre de chemin avec $k+1$ succès et commençant par un succès (c'est-à-dire commençant par S_1) au quel cas l'on complète par k succès lors de la répétition de n expériences. On obtient donc $\binom{n}{k}$ possibilités dans ce cas.
- Le nombre de chemin avec $k+1$ succès et commençant par un échec (c'est-à-dire commençant par E_1) au quel cas l'on complète par $k+1$ succès lors de la répétition de n expériences. On obtient donc $\binom{n}{k+1}$ possibilités dans ce cas.

On fait la somme du nombres de possibilités dans ces deux configurations et l'on obtient :

$$\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$$

Démonstration à connaitre absolument 2 (Formule du triangle de Pascal)

$$\begin{aligned}\binom{n}{k+1} + \binom{n}{k} &= \frac{n!}{(k+1)!(n-k-1)!} + \frac{n!}{k!(n-k)!} = \frac{n!}{k!(n-k-1)!}\left[\frac{1}{k+1} + \frac{1}{n-k}\right] \\ &= \frac{n!}{k!(n-k-1)!}\left[\frac{n-k+k+1}{(k+1)(n-p)} + \frac{1}{n-k}\right] = \frac{n! \times (n+1)}{k!(n-k-1)! \times (k+1)(n-k)} \\ &= \frac{(n+1)!}{(k+1)!(n-k)!} = \binom{n+1}{k+1}\end{aligned}$$

Démonstration à connaitre absolument 3 (Nombre de sous-ensembles : approche ensembliste)

Soit E un ensemble de cardinal fini n. Combien E admet-il de sous-ensembles ?
Pour répondre à cela posons nous la question de comment construire un sous-ensemble F de E. On prend les éléments un par un et pour chaque élément :

- soit on le "choisit"
- Soit non.

Soit 2 possibilités pour chacun des n éléments. Donc en tout, 2^n façons de construire une sous-partie de E.

Démonstration à connaitre absolument 4 (Nombre de sous-ensembles : approche calculatoire)

Soit E un ensemble fini de cardinal $n \in \mathbb{N}$. Si l'on note $\mathcal{P}_k(E)$ l'ensemble des sous-parties de E de cardinal $k \in [\![0, n]\!]$.
On a l'union disjointe :

$$\mathcal{P}(E) = \bigcup_{i=0}^{n} \mathcal{P}_k(E)$$

Donc :

$$\text{Card}(\mathcal{P}(E)) \underbrace{=}_{\text{union disjointe}} \sum_{k=0}^{n} \text{Card}(\mathcal{P}_k(E)) = \sum_{k=0}^{n} \binom{n}{k} = \underbrace{\sum_{k=0}^{n} \binom{n}{k} 1^k \times 1^{n-k}}_{\text{Binôme de Newton}} = 2^n$$

V Méthodes.

Méthode-exemple 1 (calcul des coefficients binomiaux)

On sera amenés à calculer des coefficients binomiaux "à la main".

- Retenir les formules du cours : $\binom{n}{0} = \binom{n}{0} = 1$, $\binom{n}{1} = \binom{n}{n-1} = n$, "miroir", et $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$
- Savoir construire le triangle de Pascal (et connaitre la formule de Pascal)
- Connaitre les formules :

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{n \times (n-1) \times ... \times (n-k+1)}{k \times (k-1) \times ... \times 1}$$

Pour déterminer $\binom{10}{4}$:

$$\binom{10}{4} = \frac{\overbrace{10 \times 9 \times 8 \times 7}^{\text{4 facteurs}}}{\underbrace{4 \times 3 \times 2 \times 1}_{\text{4 facteurs}}} \underbrace{=}_{\textbf{Simplifier !!!}} = \frac{10 \times 3 \times 7}{1} = 210$$

Méthode-exemple 2 (Suite de combinaisons.)

Dans le cas d'une urne avec 5 boules blanches, 3 noires et 2 rouges. On tire simultanément 5 boules.
• Le nombre de possibilités totales est simplement $\binom{10}{5}$.
• le nombre de possibilités avec 3 blanches et 2 noires est :

$$\underbrace{\binom{5}{3}}_{\text{3 B parmi 5}} \times \underbrace{\binom{3}{2}}_{\text{2 N parmi 3}} \times \underbrace{\binom{2}{0}}_{\text{0 R parmi 2}} = 10 \times 3 \times 1 = 30$$

Méthode-exemple 3 (anagramme)

Le nombre d'anagrammes de "anagramme" (attention ici on considère toutes les mots : avec ou sans sens) :

- On réfléchit en termes "d'emplacements". Il y en a 9 ici (puisque 9 lettres)
- On dénombre le nombre de représentants de chaque lettre :
  — 3 "a"
  — 2 "m"
  — 1 pour les 4 autres lettres.
- Il y a $\binom{9}{3} = 84$ façons de placer les "a".
- Il ne reste plus que $9 - 3 = 6$ emplacements possibles pour les "m" et donc $\binom{6}{2} = 15$ façons de placer les 2 "m".
- Ensuite pour le "n" : $\binom{4}{1} = 4$ (puisqu'il ne reste que 4 emplacements)
- Ensuite pour le "g" : $\binom{3}{1} = 3$
- Ensuite pour le "r" : $\binom{2}{1} = 2$
- Ensuite pour le "e" : $\binom{1}{1} = 1$

Le nombre de possibilités au total est donc :

$$\binom{9}{3} \times \binom{6}{2} \times \binom{4}{1} \times \binom{3}{1} \times \binom{2}{1} \times \binom{2}{1} = 84 \times 15 \times 4 \times 3 \times 2 \times 1$$

Méthode 4

Pour compter le nombre d'éléments vérifiant une propriété ou compter le nombre de configurations issues d'une expérience,

- on découpe le problème en plusieurs étapes
- on utilise les formules du cours pour dénombrer chaque étape.
- On fera bien attention de vérifier qu'on n'a pas oublié de cas et qu'on n'a pas compté plusieurs fois les mêmes cas (sinon il faudra diviser par le nombre de fois que l'on a compté).

VI Algorithmes.

Algorithmie 5 (liste des permutations)

```
def ieme(L,ele,i):
    G=[]
    for k in range(i):
        G.append(L[k])
    G.append(ele)
    for k in range(i,len(L)):
        G.append(L[k])
    return G

def permut(L):
    if len(L)==1:
        return [L]
    else:
        G=[[L[0]]]
        for i in range(1,len(L)):
            M=[]
            taille=len(G)
            for m in range(len(G)):
                M.append(G[m])
            for j in range(i+1):
                for k in range(taille):
                    G.append(ieme(M[k],L[i],j))
            while len(G[0])==i:
                del G[0]
        return G,len(G)
```

Algorithmie 6

Pour déterminer les 2-combinaisons et les 3-combinaisons, nous utiliserons par simplicités les listes dans Python. Alors que les listes sont des uplets et non des ensembles.

```
def deuxcombi(L):
    if len(L)<=1:
        return "aucune"
    else:
        G=[]
        for i in range(len(L)-1):
            for j in range(i+1,len(L)):
                G.append([L[i],L[j]])
        return G,len(G)

def troiscombi(L):
    if len(L)<=2:
        return "aucune"
    else:
        G=[]
        for i in range(len(L)-2):
            for j in range(i+1,len(L)-1):
                for k in range(j+1,len(L)):
                    G.append([L[i],L[j],L[k]])
        return G,len(G)
```

Algorithmie 7

Utilisation de la formule de pascal pour trouver tous les coefficient binomiaux au rang $n \in \mathbb{N}^*$.

```
def coefbin(n):
    L=[[0 for i in range(10)] for k in range(10)]
    if n==0:
        return L
    else:
        L[n][0]=1
        L[n][n]=1
        for i in range(1,n):
            L[n][i]=coefbin(n-1)[n-1][i-1]+coefbin(n-1)[n-1][i]
    return L

def bino(n):
    return coefbin(n)[n]

def binom(n):
    L=[0 for i in range(n+1)]
    for i in range(n+1):
        G=[]
        for i in range(len(L)):
            G.append(L[i])
        G[0]=1
        G[i]=1
        for j in range(1,i):
            G[j]=L[j-1]+L[j]
        L=[]
        for i in range(len(G)):
            L.append(G[i])
    return L
```