Programme de colles : semaine 9, du 25/11 au 29/11

Les nouveautés par rapport à la semaine précédente sont en bleu.

1 Sommes et produits

Reprise du programme précédent sur les calculs de sommes et de produits.

La formule de Bernoulli a été vue en DS mais pas en classe, la formule pour $\sum_{k=1}^{n} k^3$ a été vue en TD mais n'est pas à connaître par cœur obligatoirement.

Sommes doubles :

- sommes doubles rectangulaires : $\sum_{\substack{n_0 \leq i \leq n_1 \\ p_0 \leq j \leq p_1}} a_{i,j} = \sum_{i=n_0}^{n_1} \left(\sum_{j=p_0}^{p_1} a_{i,j} \right) = \sum_{j=p_0}^{p_1} \left(\sum_{i=n_0}^{n_1} a_{i,j} \right)$.
- On note $\sum_{n_0 \leq i,j \leq n_1} = \sum_{\substack{n_0 \leq i \leq n_1 \\ n_0 \leq j \leq n_1}}$
- cas des termes séparables pour les sommes rectangulaires : $\sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} a_i b_j = \left(\sum_{i=1}^{n} a_i \right) \times \left(\sum_{j=1}^{p} b_j \right)$
- sommes triangulaires : $\sum_{n_0 \leq i \leq j \leq n_1} a_{i,j} = \sum_{i=n_0}^{n_1} \left(\sum_{j=i}^{n_0} a_{i,j} \right) = \sum_{j=n_0}^{n_1} \left(\sum_{i=n_0}^{j} a_{i,j} \right)$
- choix judicieux de l'ordre de sommation : exemple du calcul de $\sum_{1 \leq i \leq j \leq n} \frac{1}{j}$
- sommes triangulaires strictes : $\sum_{n_0 \leq i < j \leq n_1} a_{i,j} = \sum_{i=n_0}^{n_1 - 1} \left(\sum_{j=i+1}^{n_1} a_{i,j} \right) = \sum_{j=n_0+1}^{n_1} \left(\sum_{i=n_0}^{j-1} a_{i,j} \right)$
- séparation d'une somme rectangulaire en sommes triangulaires : $\sum_{1 \leq i,j \leq n} a_{i,j} = \sum_{1 \leq i \leq j \leq n} a_{i,j} + \sum_{1 \leq j < i \leq n} a_{i,j}$. Exemple : calcul de $\sum_{1 \leq i,j \leq n} \min(i,j)$
- calcul de sommes doubles en Python par boucles `for` imbriquées

2 Polynômes réels

Attention, la factorisation de polynômes en utilisant des racines évidentes n'a pas encore été abordée en classe. On évitera les exercices trop abstraits.

- *on appelle polynôme toute fonction polynomiale, on note X la fonction $x \longmapsto x$*
- unicité des coefficients
- opérations : somme, produit, composition, dérivées successives
- degré. Par convention, $\deg(0) = -\infty$. Degré d'une somme, d'un produit, d'une dérivée.
- notations $\mathbb{R}[X]$ et $\mathbb{R}_n[X]$
- divisibilité et racines. Pour $P \in \mathbb{R}[X]$ et $\alpha \in \mathbb{R}$ on a : α est racine de P ssi $X - \alpha$ divise P $(*)$
- un polynôme de degré n admet au plus n racines, seul le polynôme nul admet une infinité de racines. *Les racines multiples n'ont pas encore été abordées en classe.*

3 Informatique en langage Python

Attention, les indices des éléments d'une liste n'ont pas été abordés en classe. Modification et suppression d'éléments seront également vus plus tard.

Listes :

- création par `append` successifs depuis la liste vide, création par la syntaxe `[f(k) for k in L]` où `L` est une autre liste ou un range
- parcours d'une liste par ses éléments
- fonction `sum` : *on recommande aux élèves d'utiliser `sum` lorsqu'apparaît la somme des éléments d'une liste déjà disponible, et de revenir à une boucle `for` sinon.*
- fonctions `len`, test `in`, concaténation de listes

4 Questions de cours

Les premières minutes de la colle porteront sur une ou plusieurs des questions suivantes :

1. Rappeler la définition des coefficients binomiaux puis énoncer et démontrer une formule parmi : formule de symétrie, formule d'absorption, formule de Pascal.
2. Énoncer la formule du binôme de Newton.
3. À l'aide du binôme de Newton et du triangle de Pascal, développer rapidement $(1+x)^5$ et $(x-1)^5$ pour $x \in \mathbb{R}$ (ou des quantités similaires choisies par l'examinateur).
4. Calculer $\displaystyle\sum_{1 \leq i,j \leq n} ij$.
5. Pour $a, b \in \mathbb{R}$ et $n \in \mathbb{N}$ calculer $\displaystyle\sum_{0 \leq j \leq k \leq n} \binom{n}{k}\binom{k}{j} a^j b^{k-j}$.
6. Donner la définition du degré d'un polynôme et énoncer la formule donnant le degré d'un produit.
7. Énoncer le théorème faisant le lien entre racine d'un polynôme et factorisation. *On attend le théorème $(*)$ et on s'assurera que les élèves connaissent la définition de "racine" et de "divise".*
8. Écrire une fonction Python prenant en argument un entier n et renvoyant la liste $[u_0, u_1, u_2, \dots, u_n]$ où (u_n) est une suite de réels choisie par l'examinateur du type $u_n = f(n)$ ou $u_{n+1} = f(u_n)$ ou $u_{n+1} = f(u_n, n)$.
9. Écrire une fonction Python `minimum` prenant en argument une liste de nombres réels et renvoyant le minimum de cette liste. *On admettra que le premier élément d'une liste `L` est donné par `L[0]`.*
10. Rappeler la définition de la moyenne et de l'écart type d'une série $(x_1, x_2, \dots, x_n)$ de nombres réels. Écrire deux fonctions Python prenant en argument une liste de nombres réels et renvoyant respectivement leur moyenne et leur écart-type *(cf TP 9)*.

La colle se poursuivra avec un ou plusieurs calcul "type remédiation" au sein ou non d'un exercice plus compliqué. Cette semaine, ces calculs doivent être similaires à ceux traités dans les feuilles de :

- Remédiation 5, tous les exos : `https://cahier-de-prepa.fr/bcpst1b-berthelot/download?id=5400`

La question de cours est noté sur 10 points, le reste des exercices sur 10 autres points.