Programme de colles : semaine 20, du 10/3 au 14/3

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

1 Dénombrement

Dans tout ce qui suit, E et F sont des ensembles finis.

- on note $Card(E)$ le cardinal de E
- on a $Card(E) = N \in \mathbb{N}^*$ si et seulement s'il existe une bijection $f : [\![1, N]\!] \longrightarrow E$
- E et F sont de même cardinal si et seulement s'il existe une bijection $f : E \longrightarrow F$
- cardinal d'une union : $Card(E \cup F) = Card(E) + Card(F) - Card(E \cap F)$. *Les élèves n'ont pas à connaître la formule du crible pour plus de deux ensembles, mais le cas de trois ensembles a été traité en exercice.*
- notion de partition d'un ensemble E
- si $(A_1, \ldots, A_p)$ est une partition de E alors $Card(E) = \sum_{i=1}^{p} Card(A_i)$
- cardinal du complémentaire : si $F \subset E$ alors $Card(E \backslash F) = Card(E) - Card(F)$
- si $F \subset E$ alors on a $Card(F) \leq Card(E)$, et $Card(E) = Card(F)$ si et seulement si $E = F$
- cardinal d'un produit cartésien : $Card(E \times F) = Card(E) \times Card(F)$ et $Card(E^p) = Card(E)^p$
- nombre de p-listes d'un ensemble à n éléments : n^p
- nombre de p-arrangements[1] d'un ensemble à n éléments : $\mathcal{A}_n^p = \dfrac{n!}{(n-p)!}$
- nombre de permutations d'un ensemble à n éléments : $n!$.
- nombre de p-combinaisons d'un ensemble à n éléments : $\binom{n}{p} = \dfrac{n!}{p!(n-p)!}$
- nombre de parties d'un ensemble à n éléments : 2^n
- nombre d'anagrammes d'un mot donné
- savoir reconnaître dans une situation concrète s'il s'agit de p-listes, p-arrangements, permutations, p-combinaisons, ensemble des parties
- interprétations combinatoires des formules sur les coefficients binomiaux : symétrie ($\binom{n}{k} = \binom{n}{n-k}$), formule de Pascal ($\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$), formule "du chef" ou d'absorption ($\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$), binôme de Newton

2 Informatique en langage Python

- tableaux avec la bibliothèque `numpy` :
  - création avec les commandes : `np.array`, `np.ones`, `np.eye`, `np.zeros`, `np.diag`
  - opérations coefficients par coefficients. *Nous n'avons pas vu en cours le résultat de `A==B` si `A` et `B` sont des tableaux.*
  - accès aux dimensions avec `np.size(A,0)` et `np.size(A,1)`
  - accès ou modifications des coefficients : si `A` est un tableau à n lignes et p colonnes alors on accède à ses coefficients par la syntaxe `A[i,j]` pour $i \in [\![0, n-1]\!]$ et $j \in [\![0, p-1]\!]$
  - création d'une matrice à partir de la matrice nulle via deux boucles `for` imbriquées
- tri d'une liste de nombres : tri par sélection, tri par insertion. *Seules les versions "en place" ont été présentées en classe.*

1. le programme recommande plutôt l'expression "p-listes sans répétitions"

3 Questions de cours

Cette semaine, les colles suivront le schéma suivant :

A) un calcul de primitives en utilisant la dérivée d'une composée, voir Feuille de remédiation 10, exo 2 page 4 : `https://cahier-de-prepa.fr/bcpst1b-berthelot/download?id=5713`

B) une ou deux questions de cours issue de la liste ci-dessous

C) exercices au choix du colleur

1. Énoncer la formule donnant le cardinal d'une union de 2 ensembles finis.
2. Énoncer la formule donnant le cardinal du complémentaire d'un ensemble fini.
3. Donner la définition d'une partition d'un ensemble. Quelle formule sur les cardinaux a-t-on dans ce cas ?
4. En utilisant la formule pour le cardinal d'une union de deux ensembles et la distributivité de l'union et l'intersection (cf exo 1 TD 13), montrer que pour trois ensembles finis E, F, G :

$$\begin{aligned} Card(E \cup F \cup G) &= Card(E) + Card(F) + Card(G) \\ &\quad - Card(E \cap F) - Card(F \cap G) - Card(E \cap G) + Card(E \cap F \cap G). \end{aligned}$$

5. Si $Card(E) = n$ que vaut $Card(\mathcal{P}(E))$? Démontrer cette formule en paritionnant $\mathcal{P}(E)$ selon le nombre d'éléments des sous-ensembles de E.
6. Déterminer le nombre d'anagrammes d'un mot choisi par l'examinateur. *Les élèves ne doivent pas simplement appliquer une formule toute faite, mais expliquer la manière de compter.*
7. Écrire une fonction Python `produitmat` prenant en arguments deux matrices A et B et renvoyant leur produit matriciel. Votre fonction devra gérer le cas où les tailles des matrices A et B ne sont pas compatibles.
8. Écrire une fonction Python `transpose` prenant en argument une matrice A et renvoyant sa transposée.
9. Écrire une fonction Python `tri` réalisant le tri d'une liste de nombres selon (au choix de l'examinateur) l'algorithme du tri par sélection ou du tri par insertion. Vous illustrerez les différentes étapes de l'algorithme sur une liste simple de votre choix.

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