Ordre lexicographique

On portera une attention particulière à **la rédaction** et à **la propreté** de la copie.
En particulier tout **graphe ou schéma** pertinent et correctement annoté sera **valorisé**.
Les résultats devront être mis en valeur dès que possible.

Indiquez sur votre copie le temps approximatif pris pour le devoir.

Contexte

L'objectif de ce devoir est d'étudier l'ordre lexicographique. Il s'agit d'une généralisation de l'ordre d'un dictionnaire: étant donné un ordre sur un ensemble quelconque, on peut définir par récurrence un ordre sur tous les mots (vus comme produits cartésiens) constitués d'éléments de cet ensemble.

...
ÉLITE
ÉLITISME
ÉLITISTE
ÉLIXIR
ELLE
ELLÉBORE
ELLIPSE
ELLIPSOÏDAL
ELLIPSOÏDE
ELLIPTIQUE
ELLIPTIQUEMENT
ÉLOCUTION
ÉLODÉE
ÉLOGE
ÉLOGIEUSEMENT
ÉLOGIEUX
...

---

1. Ordre lexicographique

1.I. Ordre lexicographique sur un produit

Soient $(E, \leq_E)$ et $(F, \leq_F)$ deux ensembles ordonnés.
Pour x_1 et x_2 deux éléments de E on note $x_1 <_E x_2$ ssi $x_1 \leq_E x_2$ et $x_1 \neq x_2$.
On définit un ordre $\preccurlyeq_{E\times F}$ sur $E \times F$ par

$$\forall\,((e_1,f_1),(e_2,f_2)) \in (E\times F)^2, \quad (e_1,f_1) \preccurlyeq_{E\times F} (e_2,f_2) \;\Leftrightarrow (e_1 <_E e_2 \text{ ou } (e_1 = e_2 \text{ et } f_1 \leq_F f_2))\,.$$

1.1. Montrer que $(E \times F, \preccurlyeq_{E\times F})$ est un ensemble ordonné.
1.2. Montrer que **si** $\leq_E$ est un ordre total sur E et $\leq_F$ est un ordre total sur F **alors** $\preccurlyeq_{E\times F}$ est un ordre total sur $E \times F$.
1.3. Faire un schéma illustrant l'ordre $\preccurlyeq_{E\times F}$.

1.II. Ordre lexicographique sur les mots

Soit $(E, \leq)$ un ensemble ordonné.
On définit par récurrence sur $n \in \mathbb{N}^*$ un ordre $\preccurlyeq_n$ sur E^n par

- Pour tout $(x_1, y_1) \in (E^1)^2$,
$$x_1 \preccurlyeq_1 y_1 \;\Leftrightarrow x_1 \leq y_1.$$
- Pour tout $n \geq 1$, pour tout $\big((x_k)_{k\in[\![1,n+1]\!]}, (y_k)_{k\in[\![1,n+1]\!]}\big) \in (E^{n+1})^2$,
$$(x_k)_{k\in[\![1,n+1]\!]} \preccurlyeq_{n+1} (y_k)_{k\in[\![1,n+1]\!]} \;\Leftrightarrow \big(x_1 < y_1 \text{ ou } \big(x_1 = y_1 \text{ et } (x_k)_{k\in[\![2,n+1]\!]} \preccurlyeq_n (y_k)_{k\in[\![2,n+1]\!]}\big)\big)\,.$$

1.4. Montrer que pour tout $n \in \mathbb{N}^*$, $(E^n, \preccurlyeq_n)$ est un ensemble ordonné.
1.5. Montrer que **si** $\leq$ est un ordre total sur E, **alors** pour tout $n \in \mathbb{N}^*$, $\preccurlyeq_n$ est un ordre total sur E^n.

On s'est précédemment restreint à comparer des n-uplets **de même longueur**. On peut en fait définir un ordre lexicographique sur l'ensemble des mots, c'est à dire pour des tailles finies quelconques et non nécessairement égales.
Plus précisément on pose $E^\star = \bigcup_{n\geq 1} E^n$ et on définit un ordre $\preccurlyeq_\star$ sur $E^\star$ par, pour tout $((a_1,\dots,a_p),(b_1,\dots,b_q)) \in (E^\star)^2$,

$$(a_1,\dots,a_p) \preccurlyeq_\star (b_1,\dots,b_q) \;\Leftrightarrow \begin{cases} (a_1,\dots,a_{\min(p,q)}) \prec_{\min(p,q)} (b_1,\dots,b_{\min(p,q)}) \\ \text{ou} \\ p \leq q \text{ et } (a_1,\dots,a_p) = (b_1,\dots,b_p) \end{cases}$$

On admet que $(E^\star, \preccurlyeq_\star)$ est bien un ensemble ordonné et que si $\leq$ est un ordre total sur E, alors $\preccurlyeq_\star$ est un ordre total sur $E^\star$.

1.6. On munit $\mathbb{N}$ de l'ordre usuel $\leq$.
Parmi les éléments suivants de $\mathbb{N}^\star$ (mots sur $\mathbb{N}$), ordonner ceux qui sont ordonnables par ordre lexicographique.

$$(1,2,3,6) \quad (1,2,4,1) \quad (1,2,1) \quad (1,3,0) \quad (2) \quad (1,2,3) \quad (1,2)$$

1.7. Commenter l'appellation «ordre lexicographique».

1.III. Un peu d'informatique

Les questions suivantes sont indépendantes.
On accordera une attention particulière à la présentation et l'explication du code.
On pourra utiliser au choix des fonctions itératives ou des fonctions récursives.
Aucune preuve de terminaison, de correction ou de complexité n'est demandée.

1.8. Écrire une fonction python `ordre_lexico` prenant en argument deux listes de flottants ℓ_1 et ℓ_2 et renvoyant la valeur de vérité de «$\ell_1 \preccurlyeq_\star \ell_2$» où $\preccurlyeq_\star$ est l'ordre lexicographique sur les listes de flottants. Par exemple
- `ordre_lexico([1,2,3,6],[1,2,4,1])` renvoie `True`.
- `ordre_lexico([1,3,1],[1,2,0])` renvoie `False`.
- `ordre_lexico([1,2,3],[1,2])` renvoie `False`.
- `ordre_lexico([1,1,2],[1,2])` renvoie `True`.

1.9. Écrire une fonction `liste_lexico` prenant en argument un entier n et un entier p et renvoyant la liste de tous les éléments de $[\![0,n-1]\!]^p$ triés par ordre lexicographique. Par exemple
- `liste_lexico(3,2)` renvoie `[[0,0],[0,1],[0,2],[1,0],[1,1],[1,2],[2,0],[2,1],[2,2]]`.
- `liste_lexico(3,4)` renvoie

```
[[0,0,0,0],[0,0,0,1],[0,0,0,2],[0,0,1,0],[0,0,1,1],[0,0,1,2],[0,0,2,0],[0,0,2,1],[0,0,2,2],
[0,1,0,0],[0,1,0,1],[0,1,0,2],[0,1,1,0],[0,1,1,1],[0,1,1,2],[0,1,2,0],[0,1,2,1],[0,1,2,2],
[0,2,0,0],[0,2,0,1],[0,2,0,2],[0,2,1,0],[0,2,1,1],[0,2,1,2],[0,2,2,0],[0,2,2,1],[0,2,2,2],
[1,0,0,0],[1,0,0,1],[1,0,0,2],[1,0,1,0],[1,0,1,1],[1,0,1,2],[1,0,2,0],[1,0,2,1],[1,0,2,2],
[1,1,0,0],[1,1,0,1],[1,1,0,2],[1,1,1,0],[1,1,1,1],[1,1,1,2],[1,1,2,0],[1,1,2,1],[1,1,2,2],
[1,2,0,0],[1,2,0,1],[1,2,0,2],[1,2,1,0],[1,2,1,1],[1,2,1,2],[1,2,2,0],[1,2,2,1],[1,2,2,2],
[2,0,0,0],[2,0,0,1],[2,0,0,2],[2,0,1,0],[2,0,1,1],[2,0,1,2],[2,0,2,0],[2,0,2,1],[2,0,2,2],
[2,1,0,0],[2,1,0,1],[2,1,0,2],[2,1,1,0],[2,1,1,1],[2,1,1,2],[2,1,2,0],[2,1,2,1],[2,1,2,2],
[2,2,0,0],[2,2,0,1],[2,2,0,2],[2,2,1,0],[2,2,1,1],[2,2,1,2],[2,2,2,0],[2,2,2,1],[2,2,2,2]]
```

1.10. (**Bonus**) Écrire une fonction `permutations_lexico` prenant en argument un entier n et renvoyant la liste de tous les mots de $[\![0,n-1]\!]^n$ où toutes les coordonnées sont distinctes, triés par ordre lexicographique. Par exemple
- `permutations_lexico(3)` renvoie `[[0,1,2],[0,2,1],[1,0,2],[1,2,0],[2,0,1],[2,1,0]]`
- `permutations_lexico(5)` renvoie

```
[[0,1,2,3,4],[0,1,2,4,3],[0,1,3,2,4],[0,1,3,4,2],[0,1,4,2,3],[0,1,4,3,2],[0,2,1,3,4],
[0,2,1,4,3],[0,2,3,1,4],[0,2,3,4,1],[0,2,4,1,3],[0,2,4,3,1],[0,3,1,2,4],[0,3,1,4,2],
[0,3,2,1,4],[0,3,2,4,1],[0,3,4,1,2],[0,3,4,2,1],[0,4,1,2,3],[0,4,1,3,2],[0,4,2,1,3],
[0,4,2,3,1],[0,4,3,1,2],[0,4,3,2,1],[1,0,2,3,4],[1,0,2,4,3],[1,0,3,2,4],[1,0,3,4,2],
[1,0,4,2,3],[1,0,4,3,2],[1,2,0,3,4],[1,2,0,4,3],[1,2,3,0,4],[1,2,3,4,0],[1,2,4,0,3],
[1,2,4,3,0],[1,3,0,2,4],[1,3,0,4,2],[1,3,2,0,4],[1,3,2,4,0],[1,3,4,0,2],[1,3,4,2,0],
[1,4,0,2,3],[1,4,0,3,2],[1,4,2,0,3],[1,4,2,3,0],[1,4,3,0,2],[1,4,3,2,0],[2,0,1,3,4],
[2,0,1,4,3],[2,0,3,1,4],[2,0,3,4,1],[2,0,4,1,3],[2,0,4,3,1],[2,1,0,3,4],[2,1,0,4,3],
[2,1,3,0,4],[2,1,3,4,0],[2,1,4,0,3],[2,1,4,3,0],[2,3,0,1,4],[2,3,0,4,1],[2,3,1,0,4],
[2,3,1,4,0],[2,3,4,0,1],[2,3,4,1,0],[2,4,0,1,3],[2,4,0,3,1],[2,4,1,0,3],[2,4,1,3,0],
[2,4,3,0,1],[2,4,3,1,0],[3,0,1,2,4],[3,0,1,4,2],[3,0,2,1,4],[3,0,2,4,1],[3,0,4,1,2],
[3,0,4,2,1],[3,1,0,2,4],[3,1,0,4,2],[3,1,2,0,4],[3,1,2,4,0],[3,1,4,0,2],[3,1,4,2,0],
[3,2,0,1,4],[3,2,0,4,1],[3,2,1,0,4],[3,2,1,4,0],[3,2,4,0,1],[3,2,4,1,0],[3,4,0,1,2],
[3,4,0,2,1],[3,4,1,0,2],[3,4,1,2,0],[3,4,2,0,1],[3,4,2,1,0],[4,0,1,2,3],[4,0,1,3,2],
[4,0,2,1,3],[4,0,2,3,1],[4,0,3,1,2],[4,0,3,2,1],[4,1,0,2,3],[4,1,0,3,2],[4,1,2,0,3],
[4,1,2,3,0],[4,1,3,0,2],[4,1,3,2,0],[4,2,0,1,3],[4,2,0,3,1],[4,2,1,0,3],[4,2,1,3,0],
[4,2,3,0,1],[4,2,3,1,0],[4,3,0,1,2],[4,3,0,2,1],[4,3,1,0,2],[4,3,1,2,0],[4,3,2,0,1],
[4,3,2,1,0]]
```