

Nous allons dans ce TP utiliser l'ordinateur pour explorer énumérativement certaines notions liées aux ensembles et au dénombrement.

Pour simplifier, nous allons utiliser aujourd'hui les listes (ordonnées, répétitions permises) de Python, mais on peut noter que le langage propose également une structure de données « ensembles » (non ordonnés, sans répétition).

Pour plus de détails : <https://docs.python.org/3/tutorial/datastructures.html>

### Question 1

Écrire une fonction `suites(n)` qui renvoie la liste de toutes les suites binaires de taille  $n$ . Par exemple, `suites(3)` pourrait renvoyer

[ [0,0,0], [0,0,1], [0,1,0], [0,1,1], [1,0,0], [1,0,1], [1,1,0], [1,1,1] ].

On pourra, pour obtenir une implémentation récursive élégante, exploiter le fait que

$$x \in \text{suites}(n+1) \iff x = y + [z], \quad y \in \text{suites}(n), \quad z \in \{0, 1\}.$$

### Question 2

Concevoir une fonction `parties(E)` qui prend en argument la liste des éléments d'un ensemble  $E$  et renvoie la liste de ses parties (elles-mêmes données sous forme de listes).

La liste des parties peut être renvoyée dans l'ordre de votre choix ; assurez-vous que votre fonction renvoie le bon nombre d'éléments (même dans le cas où  $E = \emptyset$ ). Par exemple, `parties([1,2,3])` pourrait renvoyer

[ [], [1], [2], [3], [1,2], [1,3], [2,3], [1,2,3] ].

N'y a-t-il pas un lien avec la question précédente ?

### Question 3

Générer la liste `parties_taille(k, E)` des parties de taille  $k$  d'un ensemble  $E$ .

Indication : vous pouvez raisonner récursivement ou alors utiliser le résultat de la question précédente. Quelle approche vous semble la meilleure ?

### Question 4

Étant donnés deux entiers naturels  $n \geq 0$  et  $p \geq 1$ , générer la listes  $E(n, p)$  des  $p$ -uplets d'entiers  $(a_1, \dots, a_p)$  tels que  $a_1 + \dots + a_p = n$  de deux façons différentes :

a) en exploitant la relation de récurrence

$$|E(n, p)| = \sum_{k=0}^n |E(k, p-1)|;$$

b) en utilisant la bijection avec les configurations de  $p-1$  cases noires parmi  $n+p-1$  cases rencontrée en TD et l'exercice précédent.

Est-ce que vos deux implémentations génèrent les  $p$ -uplets dans le même ordre ?