TD n°3 : quelques exercices sur la récurrence

Exercice 1

Soit $(a_n)_{n\geqslant 0}$ une suite de nombres réels ou complexes.
On pose $b_0 = 1$ et $b_n = \prod_{k=0}^{n-1}(1-a_k)$ pour $n \geqslant 1$. Montrer que $b_{n+1} = 1 - \sum_{k=0}^{n} a_k b_k$ pour tout n.

Exercice 2

On définit une suite $(u_n)_{n\geqslant 1}$ par $u_1 = 1$ et $u_{n+1} = 1 + \dfrac{n}{u_n}$ pour tout $n \geqslant 1$.
Montrer que $\sqrt{n} < u_n < \sqrt{n} + 1$ pour tout $n \geqslant 2$.

Exercice 3

On se donne une application f de $\mathbb{N}$ dans $\mathbb{N}$ telle que $f(n+1) \geqslant f(f(n)) + 1$ pour tout n de $\mathbb{N}$.
1. Par récurrence sur n, montrer $m \geqslant n \Rightarrow f(m) \geqslant n$ (et en particulier $f(n) \geqslant n$.)
2. En déduire que f est strictement croissante.
3. Montrer que $f(n) < n+1$ pour tout n, et en déduire l'unique solution f du problème.

Exercice 4

Pour tout n de $\mathbb{N}$, calculer $S_n = \sum_{k=0}^{n} (-1)^k k^2$.

Exercice 5

Pour tout n de $\mathbb{N}$, calculer $S_n = \sum_{k=0}^{n} \dfrac{k}{(k+1)!}$.

Exercice 6 (joli résultat)

Pour toute partie A finie non vide de $\mathbb{N}^*$, on note $p(A)$ l'inverse du produit des éléments de A.
Calculer $u_n = \sum p(A)$, où la somme est étendue aux parties non vides de $E_n = \{1, 2, \ldots, n\}$.

Exercice 7 (astucieux)

Soit n un entier supérieur ou égal à 2. On choisit $n+2$ nombres distincts dans $E_n = \{1, \ldots, 2n\}$.
Montrer que l'au au moins est égal à la somme de deux autres.
Est-ce encore vrai si on en choisit seulement $n+1$?

Exercice 8 (les questions 2 et 3 sont très difficiles !)

On pose $\begin{cases} A_0 = \{1\} \\ B_0 = \{0\} \end{cases}$ et, pour tout $n \geqslant 1$ de $\mathbb{N}$, $\begin{cases} A_n = A_{n-1} \cup \{b + 2^n,\ b \in B_{n-1}\} \\ B_n = B_{n-1} \cup \{a + 2^n,\ a \in A_{n-1}\} \end{cases}$

1. Écrire en Python deux fonctions A et B permettant de former les ensembles A_n et B_n.
   Vérifier par exemple que $\begin{cases} A_4 = \{1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31\} \\ B_4 = \{0,3,5,6,9,10,12,15,17,18,20,23,24,27,29,30\} \end{cases}$
   Toujours en Python, et avec $n = 4$, vérifier qu'on a les égalités $\sum_{a\in A_4} a^k = \sum_{b\in B_4} b^k$, pour $0 \leqslant k \leqslant 4$.
2. Pour tout n de $\mathbb{N}$, établir que A_n et B_n forment une partition de $\{0, 1, \ldots, 2^{n+1}-1\}$ en deux parties de cardinal 2^n, et qu'on a les égalités : $\sum_{a\in A_n} a^k = \sum_{b\in B_n} b^k$, pour $0 \leqslant k \leqslant n$.
3. Pour tout n de $\mathbb{N}$, calculer $d_n = \sum_{a\in A_n} a^{n+1} - \sum_{b\in B_n} b^{n+1}$.