Exercice - 18 : Mines (partiel) 2012 Emmanuel Schneider

Exercice 2, en direct pour les 10 dernières minutes :
Pour toute partie A de $[\![1,n]\!]$, on pose $\omega(\mathrm{A}) = \max(\mathrm{A}) - \min(\mathrm{A})$.
Calculer la moyenne M_n des ω et donner un équivalent de M_n quand n tend vers $+\infty$.

Solution. Posons : $\forall n \in \mathbb{N}^*, u_n = (2^n - 1)\mathrm{M}_n$: pour tout entier $n > 0$, u_n est la somme des diamètres des parties de $[\![1,n]\!]$ ($u_1 = \mathrm{M}_1 = 0, u_2 = 3\mathrm{M}_2 = 1$).
Il vient : $\forall n \in \mathbb{N}^*$,

$$\begin{aligned}
\overbrace{\text{somme des diamètres des parties de } [\![1,n+2]\!]}^{u_{n+2}} &= \overbrace{\text{somme des diamètres des parties de } [\![1,n+1]\!]}^{u_{n+1}} \\
&+ \overbrace{\text{somme des diamètres des parties de } [\![2,n+2]\!]}^{u_{n+1}} \\
&- \overbrace{\text{ceux qu'on a comptés deux fois, i.e dans } [\![2,n+1]\!]}^{u_n} \\
&+ \overbrace{\text{les parties restantes}^{1} \text{ contenant 1 et } n+2\text{, de diamètre } n+1}^{2^n(n+1)}
\end{aligned}$$

Donc :

$$\forall n \in \mathbb{N}^*, u_{n+2} - 2u_{n+1} + u_n = 2^n(n+1)$$

En posant $\forall k \in \mathbb{N}^*, v_k \overset{\text{déf}}{=} u_{k+1} - u_k$, il vient :

$$\forall k \in \mathbb{N}^*, v_{k+1} - v_k = 2^k(k+1)$$

Soit $n \in \mathbb{N}^*$; en sommant pour k allant de 1 à n :

$$\sum_{k=1}^{n} v_{k+1} - v_k = \sum_{k=1}^{n} 2^k(k+1)$$

D'où :

$$\begin{aligned}
\underbrace{v_{n+1}}_{u_{n+2}-u_{n+1}} - \underbrace{v_1}_{=u_2-u_1=1} &= \frac{\mathrm{d}}{\mathrm{d}x}\left[\sum_{k=1}^{n} x^{k+1}\right]_{x=2} \\
&= \frac{\mathrm{d}}{\mathrm{d}x}\left[x^2\frac{x^n-1}{x-1}\right]_{x=2} \\
&= \left[\frac{((n+2)x^{n+1} - 2x)(x-1) - (x^{n+2}-x^2)}{(x-1)^2}\right]_{x=2} \\
&= 2^{n+1}n
\end{aligned}$$

Et, de même, si $\mathrm{N} \geqslant 1$, en sommant[2] pour n allant de 1 à N :

$$\sum_{n=1}^{\mathrm{N}} u_{n+2} - u_{n+1} = \sum_{n=1}^{\mathrm{N}} \left(2^{n+1}n + 1\right)$$

1. elles sont de la forme $\{1, n+1\}$ union "une partie quelconque de $[2, n+1]$" : il y en a donc $|\mathcal{P}([2,n+1])| = 2^n$
2. petite erreur de logique

Soit :

$$\begin{aligned}
u_{N+2} - \underbrace{u_2}_{=1} &= 4\sum_{n=1}^{N}\left(2^{n-1}n\right) + N \\
&= 4\sum_{n=0}^{N-1}\left(2^{n}(n+1)\right) + N \\
&= 4\sum_{n=1}^{N-1}\left(2^{n}(n+1)\right) + N + 4 \\
&= 4\times 2^{N}(N-1) + N + 4 = 2^{N+2}(N-1) + N + 4
\end{aligned}$$

et :

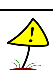
$$\forall N \geqslant 3, u_N = 2^N(N-3) + N + 3 \quad \text{(valable aussi pour N=1 et N=2)}$$

Donc[3], par définition de u :

$$\boxed{\forall n \in \mathbb{N}^*, M_n = \frac{2^n(n-3)+n+3}{2^n-1} = n-3+\frac{2n}{2^n-1} \underset{n\to\infty}{\sim} n}$$

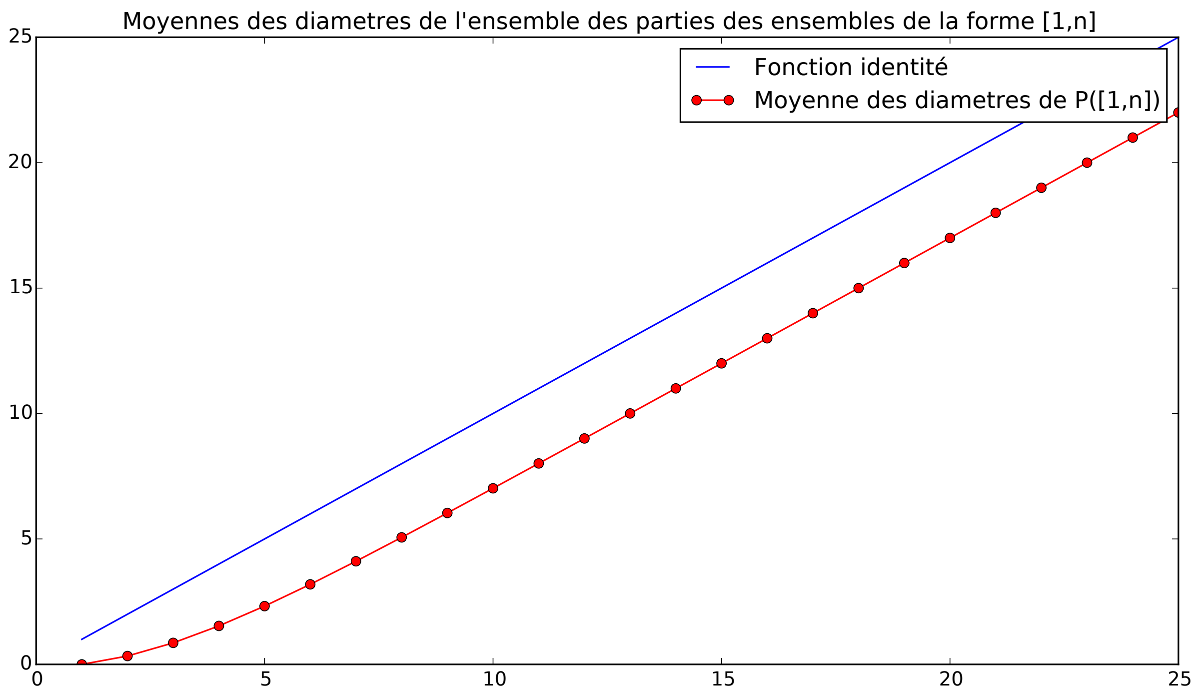
Simulation python :

```
import matplotlib.pyplot as plt
import numpy as np
def moyennes_diametres(m):
    Y=[]
    X = range(1,m+1)
    diametres = []
    for n in X:
        for i in range(2**(n-1),2**n):
            min_courant = n-1
            max_courant = 0
            for j in range(n):
                if (i>>j)&1 == 1: # on "deplace" vers la droite de j rang la
                # representation binaire de i, puis on applique l'operateur
                # binaire "AND" pour verifier si il y a un "1" en position j
                # (position 1 = unites)

                    if j < min_courant :
                        min_courant = j
                    elif j > max_courant:
                        max_courant = j
            diametres.append(max_courant - min_courant)

        Y.append(sum(diametres)/(2**n-1))

        p2 = plt.plot(X,X,label='Fonction identite')
        p1 = plt.plot(X,Y,marker='o'
                , label='Moyenne des diametres de P([1,n])'
                , color='red')%
    plt.title("Moyennes des diametres de l'ensemble des parties"
                "des ensembles de la forme [1,n]")
    plt.legend()
    plt.show()
```

3. 

```
moyennes_diametres(25)
```

□



L'écart des altitudes entre la courbe bleue et la courbe rouge, qui tend vers 3, vaut exactement : $3 - \dfrac{2n}{2^n - 1}$.