

Problème 173 – Le Jerusalema dance challenge

Niveau : Terminale (Spécialité Maths)

Chapitres : Suites numériques (Récurrence), Algorithmique (Python)

Inédit, publié le 18/11/2020



Sorti fin décembre 2019, « Jerusalema » a créé une surprise en devenant en quelques mois un véritable tube planétaire ! Initialement, cette chanson, qui est une prière, n'était pourtant destinée à être utilisée que dans les églises d'Afrique du Sud. C'était sans compter le levier de milliers d'utilisateurs de TikTok, qui ont relayé le #JerusalemaDanceChallenge durant la période du premier confinement au printemps 2020. Dans ce problème, nous allons proposer une modélisation simple et totalement imaginaire de ce dance challenge, pour justement voir comment il se développe sur une piste de danse.

La danse de « Jerusalema » est une danse du type « Madison », où l'on exécute, en mangeant avec une assiette à la main, une série de pas que l'on répète sans arrêt après avoir effectué à chaque fois $\frac{1}{4}$ de tour – ce qui signifie qu'on fait un tour entier après avoir effectué la série de pas 4 fois de suite. On suppose qu'au début de la chanson, un certain nombre de personnes se lancent sur la piste de danse. Au bout du premier tour de danse, le nombre de danseurs a doublé. Durant le deuxième tour, un nombre de personnes égal aux nombres de danseurs déjà présents sur la piste au début de ce tour rejoignent la piste de danse, mais une personne s'arrête de danser. Puis, de manière plus générale, on considère que durant le $(n+1)$ ème tour, un nombre de personnes égal au nombre de danseurs déjà présents au début de ce $(n+1)$ ème tour rejoignent la piste de danse, mais n personnes s'arrêtent de danser. On admet, dans ce scénario fictif, que la chanson tourne en boucle sans s'arrêter et que la piste de danse n'a pas de limite de taille.

On appelle (u_n) la suite qui à n associe le nombre de danseurs présents sur la piste au bout du $n^{\text{ème}}$ tour de danse, en appelant u_0 le nombre de danseurs sur la piste au début de la chanson.

1) Expliquer pourquoi pour tout $n \geq 0$, $u_{n+1} = 2u_n - n$.

2) Dans cette question uniquement, on suppose que $u_0 = 1$.

a) Calculer u_1, u_2, u_3 .

b) Conjecturer l'expression de u_n en fonction de n pour tout $n \geq 0$, et montrer cette expression par récurrence.

3) On suppose à partir d'ici que u_0 n'est pas connu. On cherche une expression explicite de u_n en fonction de n et de u_0 .

On pose pour tout $n \geq 0$:

$$\begin{cases} v_n = \sum_{k=0}^n w_k \\ \text{où } w_k = 2^k - 1 \end{cases}$$

a) Démontrer que pour tout $n \geq 0$: $v_n = 2(2^n - 1) - n$

b) Montrer par récurrence que, pour tout $n \geq 1$, $u_n = 2^n u_0 - v_{n-1}$

4) On cherche à écrire un algorithme en langage Python, qui permet de déterminer, dans ce modèle, au bout de combien de tours le nombre de personnes sur la piste de danse dépasse 10 000 personnes - étant donné un nombre initial u_0 rentré par l'utilisateur.

a) Remplir le programme ci-dessous afin qu'il puisse donner une réponse à cette question.

```
1 u=int(input("Nombre de personnes?"))
2 n=.....
3 while u<.....:
4     u=.....
5     n=.....
6 print(...)
```

b) Déterminer ce qu'affiche le programme si l'utilisateur entre la valeur 2 à l'exécution du programme.