

Mon précieux ... le retour

1 le background

Quand on manipule des suites, on a parfois des surprises dès lors que l'on modifie les conditions d'initialisation de la suite. Généralement, on ne s'attend pas à ce que la suite se comporte de façon très différente. Mais c'est parfois le cas, comme par exemple avec les suites permettant de dessiner des fractales.

Qu'en est-il dans le cas de la suite utilisée dans le problème « Mon précieux » ?

2 l'énoncé

Le contexte est le même que dans l'exercice de base mais on construit la liste u en utilisant un procédé légèrement différent. On commence par initialiser les n premiers termes u_0, \dots, u_{n-1} , avec des valeurs toutes différentes, et pas forcément ordonnées. Ensuite, on détermine u_n comme étant la valeur minimale telle que :

- $u_n > \max_{j < n}(u_j)$
- u_n peut être obtenu d'une et une seule façon en additionnant 2 valeurs u_j et u_k , avec $j \neq k$, $0 \leq j, k < n$.

Enfin, pour $i > n$, on détermine u_i exactement comme dans l'exercice de base, à savoir :

- $u_i > u_{i-1}$
- u_i peut être obtenu d'une et une seule façon en additionnant 2 valeurs u_j et u_k , avec $j \neq k$, $0 \leq j, k < i$.

Par exemple, avec $u_0 = 1$ et $u_1 = 5$, $u_2 = 2$, on obtient :

- $n = 3$. On cherche d'abord la valeur de u_3 minimum supérieure à $\max(u_0, u_1, u_2) = 5$
- $u_3 = 6$ car l'addition donnant la valeur minimum supérieure à 5 est $u_0 + u_1 = 6$,
- $u_4 = 8$ car le plus petit nombre supérieur à u_3 pour lequel il n'existe qu'une seule addition pour l'obtenir est 8, avec $u_3 + u_2$. On peut obtenir 7, mais pour cela, il y a deux additions possibles : $u_0 + u_3$ et $u_1 + u_2$.
- $u_5 = 9$,
- etc.

En revanche, la finalité de cet exercice est différente car il s'agit de déterminer si une valeur fait partie des u_i ou non.

Pour résoudre ce problème, votre programme doit lire sur l'entrée standard :

1. une ligne contenant n valeurs séparées par des espaces, représentant les valeurs u_0, \dots, u_{n-1} de la suite.
2. un entier M représentant le nombre de valeurs à tester
3. M lignes avec un entier.

Votre programme doit écrire sur la sortie standard M lignes. La i^{eme} ligne doit afficher **yes** si le i^{eme} entier fait partie de la suite, et **no** sinon.

Exemple d'entrée/sortie :

entrée	sortie
1 5 2	yes
4	no
13	yes
90	no
997	
832	

TABLE 1 – Exemple d'entrée et la sortie attendue

Commentaires :

- Il y a trois valeurs pour initialiser la suite : $u_0 = 1, u_1 = 5, u_2 = 2$,
- Il y a 4 entiers à vérifier, s'ils sont dans la liste : 13, 90, 997, 832.
- En calculant suffisamment de termes de la suite pour arriver à une valeur > 997 , on peut ensuite déterminer que 13 et 997 font partie de la suite alors que 90 et 832 non, d'où la sortie indiquée.
- etc.

IMPORTANT : le temps limite pour trouver la solution est de 8 secondes. Si votre programme n'est pas capable de calculer les solutions du deuxième fichier de test fourni (cf. ci-dessous) en moins de 8 secondes, il sera considéré en échec et n'arrivera pas non plus à passer le plus gros fichier secret.

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur <http://cours-info.iut-bm.univ-fcomte.fr> un article dans la section **hackaton** de l'année courante, portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code, ainsi que des fichiers d'entrée et sortie pour tester votre programme.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.