PROJET EULER
Problème 119

Table des matières

1 Rappels de nos objectifs

Suite à notre dernier entretien, nos objectifs sont les suivants :
- Etudier la piste du logarithme afin de déterminer à l'avance nos bornes de recherche
- Déterminer les limites de notre fontion qui calcule la somme des chiffres d'un nombre
- Essayer de supprimer ce phénomène de chance dans notre programme

2 Comment trouver les bornes de recherche ?

Ici, notre objectif est de borner a et b de manière à ce que a^b soit inférieur à un certain nombre défini à l'avance. Ainsi, on considère que lorsque l'on cherche le n-ième terme de notre liste de nombres, on connait approximativement son ordre de grandeur et on peut trouver une valeur supérieure à cette dernière. Si on appelle "vMax" cette valeur maximum, le but est de donc de trouver pour quelles valeurs de a et de b, $a^b < vMax$.

2.1 Comment borner a ?

Nous devons trouver la valeur maximum de a telle que $a^b < vMax$. Pour cela, il faut prendre la valeur minimum de b, car $\forall a1, a2, vMax, b1, b2 \in \mathbb{R}, b1 > b2 \Rightarrow \exists a1 > a2, a2^{b1} > a1^{b2}$.

Ainsi, on obtient l'inéquation suivante :

$$a^2 < vMax$$

$$\Leftrightarrow \ln a^2 < \ln vMax$$

$$\Leftrightarrow 2 * \ln(a) < \ln(vMax)$$

$$\Leftrightarrow \ln(a) < \frac{\ln(vMax)}{2}$$

$$\Leftrightarrow a < e^{\frac{\ln(vMax)}{2}}$$

Ainsi, si l'on connaît vMax, on sait à l'avance que a ne devra pas dépasser $e^{\frac{\ln(vMax)}{2}}$

2.2 Comment borner b ?

Etant donné que l'on connaît la valeur de a avant de déterminer la valeur de b, on peut exprimer la valeur limite de b en fonction de a et de vMax. On a donc l'inéquation suivante :

$$a^b < vMax$$

$$\Leftrightarrow \ln a^b < \ln vMax$$

$$\Leftrightarrow b * \ln(a) < \ln(vMax)$$

$$\Leftrightarrow b < \frac{\ln(vMax)}{\ln a}$$

Ainsi, si l'on connaît vMax et a , on sait à l'avance que b ne devra pas dépasser $\frac{\ln(vMax)}{\ln a}$

3 Optimisation de la fonction qui calcule la somme des chiffres d'un nombre ?

Une fois que l'on avait réussi à borner a et b, notre temps de calcul augmenta largement car en supprimant la chance de notre programme on augmente largement le nombre d'appels à la fonction sommeChiffresNombre afin d'être sûr de trouver le bon résultat. Nous avons donc essayé de réduire le temps de calcul de cette fonction, d'autant plus que celle-ci ne fonctionne plus à partir d'une certaine valeur. En effet, nous n'avons pas réussi à en connaître la raison mais à partir de 10^{17} notre fonction ne retourne pas le bon résultat. Nous avons donc essayé une différente méthode pour calculer la somme des chiffres d'un nombre et cette dernière s'est retrouvé beaucoup plus efficace, pas tellement au niveau du temps de calcul mais plus au niveau de la complexité et du fait qu'elle fonctionne pour toutes les valeurs. Elle consiste à transformer le nombre en chaîne de caractères et calculer la somme des caractères de la chaîne obtenue. En python, la fonction correspond donc à :

```
1 def sommeChiffreNombre(nombre):
2     return sum([ int(c) for c in str(nombre) ])
```

4 Comment déterminer vMax ?

Pour pouvoir borner a et b, il faut d'abord déterminer vMax. Or, si l'on en choisit un trop petit, le programme retourne une erreur puisqu'il ne trouve pas assez de valeurs et s'il est trop grand, il devient beaucoup trop long puisqu'il trouve plus de valeurs que demandées. On se retrouve donc dans une situation où il faut avoir de la chance pour trouver le bon résultat. Nous avons donc modifié notre programme de façon que si l'on rentre un vMax trop petit, ce dernier augmente tout seul jusqu'à ce qu'il trouve la bonne valeur. De cette façon, en rentrant un vMax très petit, on est certain que le programme nous renverra le bon résultat mais par contre cela risque d'être très long. On se retrouve donc devant un choix. Doit-on privilégier le temps d'exécution ou la certitude ?