

PROJET EULER

Problème 119

Table des matières

1	Présentation du problème	1
2	Méthodes de réflexion	1
2.1	Méthode itérative	1
2.2	Seconde méthode	2
3	Le problème du "J'ai de la chance"	2
4	Résultat de notre entretien	2

1 Présentation du problème

Le problème 119 du Projet Euler consiste à trouver le 30ème nombre tel que la somme de ses chiffres élevé à une certaine puissance soit égale à lui-même. Par exemple, le 1er nombre respectant ces règles est 81 car la somme des chiffres de 81 est : $8 + 1 = 9$ et $9^2 = 81$. De même, le deuxième est 521 : $5 + 2 + 1 = 8$ et $8^3 = 521$. Dans l'énoncé du problème, on nous dit que le dixième est 614 656. On rappelle qu'un nombre doit contenir au moins deux chiffres pour avoir une somme.

2 Méthodes de réflexion

2.1 Méthode itérative

Au tout début, on a commencé à réfléchir à comment on pouvait calculer la somme des chiffres d'un nombre. Notre méthode consiste à récupérer le reste de la division du nombre par 10, de l'ajouter à la somme puis faire la division entière de notre nombre par 10 et répéter ceci jusqu'à ce qu'il soit égal à 0. Par exemple, pour 614 656, le reste de $614656/10$ est 6 et la division entière de 614 656 par 10 donne 61 465. Ensuite le reste de $61465/10$ est 5. Ainsi de suite, on obtient $6 + 5 + 6 + 4 + 1 + 6$. Lorsqu'on en arrive là, on divise 6 par 10, ce qui donne 0 et qui cause la sortie de la boucle. On a donc obtenu la somme des chiffres d'un nombre.

Ensuite, on a commencé à vérifier pour tous les nombres en partant de 10 s'il fait partie des nombres que l'on recherche. Ainsi, pour chaque nombre, on calcule la somme de ses chiffres, on l'élève au carré (on appellera ce nombre n), on vérifie si le résultat est égal au nombre de départ, auquel cas on ajoute 1 à un compteur qui doit atteindre 30. Si n est inférieur au nombre de départ, on augmente la puissance et on recommence, s'il est supérieur on s'arrête car il ne pourra

plus jamais être égal au nombre de départ, et on passe au nombre suivant. Lorsque le compteur atteint 30 c'est qu'on a trouvé le nombre que l'on cherche. Or, cette méthode est beaucoup trop longue car le 30ème est un nombre très très grand ce qui veut dire qu'on calcule la somme des chiffres de tous les nombres avant de trouver le bon. D'après nos calculs, la complexité en temps de cette méthode est en $O(e^n)$ (j'ai mis une complexité au pif pour l'instant) où n est le nombre de chiffres du nombre que l'on recherche.

2.2 Seconde méthode

Il a donc fallu que l'on trouve une méthode plus rapide. Nous n'avons pas essayé de réduire la complexité de notre fonction qui calcule la somme des chiffres d'un nombre mais plutôt d'atteindre plus rapidement le 30ème sans avoir à calculer la somme des chiffres de tous les nombres. Ainsi, nous nous sommes dit que l'on pouvait prendre des petits nombres " a " (par exemple jusqu'à 400 pour commencer) que l'on élève à des petites puissances " b " (par exemple jusqu'à 50 pour l'instant) et calculer la somme des chiffres de chacun de ces nombres " a^b ". Lorsque cette dernière est égale à " a ", on ajoute ce dernier à une liste. A la fin, on trie la liste pour en récupérer le 30ème. De cette façon, on calcule la somme des chiffres de beaucoup moins de nombre puisqu'il y en a une multitude qui ne sont pas une puissance d'un autre. Par exemple, aucun nombre élevé à une certaine puissance est égal à 26 donc on ne calcule jamais la somme des chiffres de 26. Ceci réduit clairement la complexité en temps de notre programme et nous avons réussi à trouver le nombre recherché qui est 248 155 780 267 521.

3 Le problème du "J'ai de la chance"

Malheureusement, cette méthode a posé un autre problème. Si l'on veut développer un programme générique qui pourrait retourner n'importe lequel nombre de la liste de ceux qui respectent les règles, on se rend compte qu'il faut compter sur la chance. En effet, dans notre cas, nous avons décidé d'aller jusqu'à 400^{50} et heureusement nous avons trouvé le bon résultat. Mais rien ne nous disait auparavant que le nombre que l'on cherchait n'était pas égal à 500^{10} ou 5^{75} (sachant que ces deux nombres sont plus petits que 400^{50}). Si c'était le cas, notre programme n'aurait pas renvoyé le bon résultat puisqu'il ne l'aurait même pas étudié donc pas ajouté à la liste. C'est ainsi qu'on s'est demandé s'il n'y avait pas un moyen de connaître à l'avance le " a " et le " b " maximum pour lesquels le programme donne le bon résultat.

4 Résultat de notre entretien

Suite à notre entretien, nos objectifs sont les suivants :

- Déterminer les limites de notre fonction qui calcule la somme des chiffres d'un nombre.
- Etudier la piste du logarithme afin de déterminer à l'avance nos bornes de recherche
- Essayer de supprimer ce phénomène de chance dans notre programme