A1857. Allergiques à la primalité

Fabien PETITJEAN

1. Trouver un nombre entier composé N_1 de dix chiffres qui reste composé quand on change l'un quelconque de ses dix chiffres par un autre chiffre. Montrer qu'il existe une infinité de nombres entiers composés qui ont la même propriété que N_1.
2. Trouver le nombre entier composé N_2 à deux chiffres en représentation décimale qui reste composé quand on change un ou deux chiffres de sa représentation binaire. Il est permis de changer en 0 le ou les chiffre(s) le(s) plus significatif(s).

1 Réponse à la question 1

Si on choisit un multiple de 10 pour N_1, on voit clairement que la modification de tout chiffre qui n'est pas l'unité donne un nombre qui est toujours multiple de 10. Quant au chiffre de l'unité, le remplacer par un chiffre pair donne un nombre pair. Pour qu'il reste composé quand on remplace le zéro de l'unité par un 3 ou un 9, il suffit qu'il soit multiple de 3. De même, pour le chiffre 7, il faut que N_1 soit multiple de 7. Au final, N_1 doit être multiple de $10 \times 3 \times 7 = 210$.

Il nous reste à trouver un moyen de d'obtenir un nombre composé en changeant le dernier chiffre par un 1. Autrement dit, il faut factoriser $N_1 + 1$.

Pour cela, on remarque que $(a+1)^2 = a^2 + 2a + 1 = a(a+2)+1$. Si N_1 est de la forme $a(a+2)$ notre problème est résolu. Et puisque N_1 est multiple de 210, il suffit de choisir un entier naturel n tel que $N_1 = 210n(210n+2)$.

Il existe donc une infinité de nombres composés qui restent composés quand on change l'un de ses chiffres.

Pour trouver un exemple numérique à 10 chiffres, il faut remarquer que $210n(210n+2) > 210^2n^2 > 10^4n^2$. Cela signifie que si n a plus de 3 chiffres, N_1 aura plus de 10 chiffres. Posons $f(n) = N_1$. C'est une fonction croissante, on peut donc appliquer une dichotomie sur l'intervalle $[0, 1000]$.

$f(1000) = 44100420000$, $f(500) = 11025210000$, $f(250) = 2756355000$

Le nombre $\boxed{\mathbf{2756355000}}$ convient et on peut vérifier que $2756355000 + 1 = 52501^2$.

2 Réponse à la question 2

Un nombre à deux chiffres (en base 10) est constitué d'au plus 7 bits. Il y a $6 + 5 + \cdots + 1 = 15$ façons de modifier deux de ses bits. Et si on modifie un seul bit, la parité n'est inversée qu'avec le bit des unités. Il y a donc assez peu de cas à tester. On pourrait le faire à la main avec un peu de patience, mais l'algorithme suivant le fait aussi bien et nous donne l'unique solutions $\boxed{84_{1010100}}$.

```
primes = [2, 3, 5, 7, 11, 13, 17, 19, 23,
    29, 31, 37, 41, 43, 47, 53, 59, 61,
    67, 71, 73, 79, 83, 89, 97]

def ok(n):
    if n + 1 in primes:
        return False
    for a in range(1, 7):
        maskA = 2**a
        if n >= maskA:
            for b in range(a):
                maskB = 2**b
                v = n ^ maskA ^ maskB
                if v in primes:
                    return False
    return True

for n in range(10, 101):
    if ok(n): print(n, bin(n))
```