TIPE : Miller-Rabin prime test

1 Principe

1.1 Propriété / définition (*témoin de Miller-Rabin*)

Soit $p \in \mathbb{N} \mid p > 2$ premier.
Soient $s, d \in \mathbb{N} \mid p - 1 = 2^s d$, avec $d \equiv 1\ [2]$.
Considérons $a \in \mathbb{N} \mid a \wedge p = 1$, appelé *base*.
Alors

$$\begin{cases} a^d \equiv 1\ [p] \\ \text{or} \\ \exists r \in [\![0\ ;\ s-1]\!]\ \mid a^{2^r d} \equiv -1\ [p] \end{cases}$$

En effet, d'après le petit théorème de Fermat, on a

$$a^{p-1} \equiv \left(a^d\right)^{2^s} \equiv 1\ [p]$$

Or comme p est premier, $\mathbb{Z}/p\mathbb{Z}$ est un corps, donc l'équation $X^2 = 1$ n'a que deux solutions dans ce corps : ± 1. Donc en prenant les racines carrées de a^{p-1} de façon répétée, soit on obtient toujours 1, soit on obtient à un moment -1.

Donc par contraposition :

> Pour $n \in \mathbb{N},\ n > 2,\ s, d \in \mathbb{N} \left| \begin{array}{l} n - 1 = 2^s d \\ d \equiv 1\ [2] \end{array} \right.$, $a \in [\![2\ ;\ n-1]\!]$, si
>
> $$\begin{cases} a^d \not\equiv 1\ [n] \\ \forall r \in [\![0\ ;\ s-1]\!],\ a^{2^r d} \not\equiv -1\ [n] \end{cases}$$
>
> alors n est composé (*i.e* n'est pas premier).
> Dans ce cas, on dit que a est un *témoin de Miller*

Dans le cas contraire, on dit que n est *fortement probablement premier en base a*. Mais n n'est pas forcément premier.

Si n est fortement probablement premier en base a mais n'est pas pourtant premier, on dit que n est un *menteur fort*.

1.2 Propriété

> Soit $n \in \mathbb{N} \left| \begin{array}{l} n > 2 \\ n \equiv 1\ [2] \\ n \notin \mathbb{P} \end{array} \right.$
> Alors au moins $\dfrac{3}{4}$ des entiers de $[\![2\ ;\ n-1]\!]$ sont des témoins de Miller pour n.

Il y a donc toujours au moins un témoin de Miller pour un nombre composé impair, donc l'équivalent des nombres de Carmichael n'existe pas pour le test de Miller-Rabin.

De plus, si l'on fait ce test avec d'autres bases, on diminue la probabilité qu'un entier composé soit déclaré comme premier.

1.3 Remarques

Si le résultat de ce test pour un entier n dit que ce nombre est composé, alors n est forcément composé.

Sinon, n est probablement premier, avec la probabilité $\left(\frac{3}{4}\right)^k$, où k est le nombre de tests.

2 Code

```python
##-Probabilistic prime test
def isSurelyPrime(n):
    '''Check if n is probably prime. Uses Miller Rabin test.'''

    if n == 2:
        return True

    elif n % 2 == 0:
        return False

    return miller_rabin(n, 15)


def miller_rabin_witness(a, d, s, n):
    '''
    Return True if a is a Miller-Rabin witness.

    - a : the base ;
    - d : odd integer verifying n - 1 = 2^s d ;
    - s : positive integer verifying n - 1 = 2^s d ;
    - n : the odd integer to test primality.
    '''

    r = pow(a, d, n)

    if r == 1 or r == n - 1:
        return False

    for k in range(s):
        r = r**2 % n

        if r == n - 1:
            return False

    return True

```

```python
def miller_rabin(n, k=15) :
    '''
    Return the primality of n using Miller-Rabin probabilistic primality
    test.

    - n : odd integer to test the primality ;
    - k : number of tests (Error = 4^(-k)).
    '''

    if n in (0, 1):
        return False

    if n == 2:
        return True

    s, d = max_parity(n - 1)

    for i in range(k) :
        a = randint(2, n - 1)

        if miller_rabin_witness(a, d, s, n):
            return False

    return True
```