

Ce petit code ci-dessous en bleu donne une méthode de calcul de multiplication plus de 2 fois plus rapide que gmp !!!!!

Testez le sous python avec la library gmpy2 et si vous trouvez ça étonnant reconnaissez mon travail et faites le connaitre autour de vous.

David Strainchamps

```
import gmpy2
from gmpy2 import mpz

import time
import timeit
import statistics

def carre(x):
    n1=gmpy2.divexact(x,2)
    X=gmpy2.add(n1**2,n1)
    return gmpy2.add(1,gmpy2.mul(X,4))

mysetup="""
import gmpy2
from gmpy2 import mpz
import numpy as np
"""

mycode="""
n =10000000
rs = gmpy2.random_state(hash(gmpy2.random_state()))
X1 = gmpy2.mpz_urandomb(rs, n)
n =100000
rs = gmpy2.random_state(hash(gmpy2.random_state()))
X2 = gmpy2.mpz_urandomb(rs, n)
Somme=gmpy2.add(X1,X2)
Differ=gmpy2.sub(X1,X2)
def carre(x):
    n1=gmpy2.div(x,2)
    X=gmpy2.add(n1**2,n1)
    return gmpy2.add(1,gmpy2.mul(X,4))
if (gmpy2.is_odd(Somme)):
    carre2=carre(Somme)
else:
    cpt=0
    while(gmpy2.is_even(Somme)):
        Somme=gmpy2.divexact(Somme,2)
        cpt=cpt+1
    carre2=gmpy2.mul(carre(Somme),mpz('4')**cpt)
```

```

if (gmpy2.is_odd(Differ)):
    carre1=carre(Differ)
else:
    cpt=0
    while(gmpy2.is_even(Differ)):
        Differ=gmpy2.divexact(Differ,2)
        cpt=cpt+1
    carre1=gmpy2.mul(carre(Differ),mpz('4')**cpt)
produit=gmpy2.sub(carre1,carre2)
"""

print(timeit.repeat(setup = mysetup,stmt=mycode,number=100))

```

```

mycode2="""
n =10000000
rs = gmpy2.random_state(hash(gmpy2.random_state()))
X1 = gmpy2.mpz_urandomb(rs, n)
n =100000
rs = gmpy2.random_state(hash(gmpy2.random_state()))
X2 = gmpy2.mpz_urandomb(rs, n)
produit=gmpy2.mul(X1,X2)
"""

print(timeit.repeat(setup = mysetup,stmt=mycode2,number=100))

```