

# Complexité expérimentale : un exemple

En python les entiers sont de taille quelconque (non bornée). On va mesurer le temps nécessaire pour effectuer des sommes et des produits d'entiers, avec des nombres de taille variable, et afficher les temps d'exécution relatifs en fonction de leur tailles (mésurées par le nombre de digits en base 10).

On réalise ceci dans une fonction `test(N,repeat)` où l'on fait varier le nombre de chiffres de deux entiers `x` et `y` de 6 chiffres à `N` chiffres. On répète la somme et le produit de `x` et `y` un grand nombre de fois de fois, le nombre de répétitions étant le paramètre `repet` de la fonction, pour gommer les légères différences en temps et améliorer la précision de la mesure. On met les temps de calcul dans deux listes, `somme` et `produit`, pour les représenter graphiquement à la fin.

Entrée [1]:

```
def test(N,repeat):
    from timeit import default_timer as timer

    c = 6 # --- on va commencer avec des nombres de 6 chiffres
    x = 123456 # --- 2 entiers x et y
    y = 987654 # de 6 chiffres
    somme = []
    produit = []
    #
    for n in range(6,N + 1): # --- on va varier le nombre de chiffres
                             # de 6 à N
        t0 = timer()
        for h in range(repet): # --- on refait repet fois
            z = x + y
            delta = timer() - t0
            somme.append(delta)
            #
            t0 = timer()
            for h in range(repet): # --- on refait repet fois
                z = x*y
                delta = timer() - t0
                produit.append(delta)
                #
            x = 10*x + 1 # --- x et y ont un
            y = 10*y + 1 # chiffre de plus
    return somme,produit
```

Dans ce qui suit, on va mesurer le temps de d'exécution de  $r = 10^3$  opérations (identiques) avec chaque paire de valeurs pour `x` et `y`, dont le nombre de chiffres va varier de 6 à  $N = 100$ .

Entrée [2]:

```
N = 100
r = 1000
s,p = test(N,r)
```

Entrée [3]:

```
# --- on imprime les temps d'exécution pour la somme
print("\nPour la somme (n et temps d'exéc. pour n chiffres) :\n")
print("N \t      temps en secs")
print("-----")
for n in range(6,N+1):
    print(n, "\t",s[n-6])
```

Pour la somme (n et temps d'exéc. pour n chiffres) :

N	temps en secs
6	4.8144000000008323e-05
7	4.7333000000003826e-05
8	4.85880000000071574e-05
9	4.787699999997397e-05
10	4.98859999999899e-05
11	5.012199999998579e-05
12	5.00549999999933e-05
13	4.9522000000007956e-05
14	4.9509000000003134e-05
15	4.94410000000038926e-05
16	4.957499999991288e-05
17	4.940700000000131e-05
18	4.9158000000004911e-05
19	5.050799999994915e-05
20	5.025899999996945e-05
21	5.024599999992052e-05
22	5.01270000000066424e-05
23	5.0604000000000964e-05
24	5.02890000000036665e-05
25	5.1470000000005342e-05
26	5.0144000000002972e-05
27	5.00750000000093794e-05
28	5.96870000000030466e-05
29	5.1765000000003692e-05
30	5.220599999999086e-05
31	5.08620000000040375e-05
32	5.0651999999984376e-05
33	5.084999999993567e-05
34	5.1307000000000436e-05
35	5.0567000000005687e-05
36	5.1161000000002177e-05
37	5.0935999999945913e-05
38	5.19930000000027934e-05
39	5.2950000000005157e-05
40	5.1996000000005411e-05
41	5.132499999993545e-05
42	5.1432999999989626e-05
43	5.22829999999308e-05
44	5.176400000000652e-05
45	5.255999999997929e-05
46	5.3744999999993937e-05
47	5.5416000000003033e-05
48	5.343499999999057e-05
49	5.3744000000002222e-05
50	5.32720000000048614e-05
51	5.456599999997813e-05
52	5.4026000000004032e-05

53	5.270800000001241e-05
54	5.354800000001436e-05
55	5.460499999998536e-05
56	5.4029000000066496e-05
57	5.427499999992524e-05
58	5.491600000007146e-05
59	5.4038000000034e-05
60	7.005199999998712e-05
61	5.408299999998256e-05
62	5.36509999999879e-05
63	5.445800000003498e-05
64	5.533999999995931e-05
65	5.504300000003237e-05
66	5.5731000000003306e-05
67	5.650000000001487e-05
68	5.571299999995727e-05
69	5.5387999999934046e-05
70	5.8809999999964724e-05
71	5.547299999997257e-05
72	5.6774000000037184e-05
73	0.00016194300000005324
74	5.714199999995451e-05
75	6.152900000000461e-05
76	5.9048000000005985e-05
77	5.571900000000962e-05
78	5.57019999999353e-05
79	5.66699999999809e-05
80	5.5978000000012074e-05
81	5.8149000000007334e-05
82	5.842000000000347e-05
83	5.7776000000009375e-05
84	5.631500000002898e-05
85	5.533799999990485e-05
86	5.545200000001138e-05
87	5.484300000002662e-05
88	5.647100000005789e-05
89	5.5084999999954753e-05
90	5.500599999996858e-05
91	5.566700000003699e-05
92	5.551999999997559e-05
93	5.524999999995117e-05
94	5.572499999995095e-05
95	5.527999999999089e-05
96	5.545200000001138e-05
97	5.676700000001311e-05
98	5.471399999990023e-05
99	5.478100000000374e-05
100	5.6692000000024834e-05

Entrée [4]:

```
# --- on imprime les temps d'exécution pour le produit
print("\nPour le produit (n et temps d'exéc. pour n chiffres) :\n")
print("N \t      temps en secs")
print("-----")
for n in range(6,N+1):
    print(n, "\t", p[n-6])
```

Pour le produit (n et temps d'exéc. pour n chiffres) :

N	temps en secs
6	4.9151999999996754e-05
7	4.89499999999936544e-05
8	5.2461999999997536e-05
9	5.12619999999940855e-05
10	7.0468999999996192e-05
11	7.1505999999994345e-05
12	7.0256999999996249e-05
13	6.9869000000005569e-05
14	6.9930000000010685e-05
15	6.9648000000008875e-05
16	6.9955999999992663e-05
17	7.0009999999995372e-05
18	6.961999999999247e-05
19	7.9366000000002514e-05
20	8.0268000000004997e-05
21	8.004999999999818e-05
22	8.0020000000006948e-05
23	7.9829000000003124e-05
24	8.0813999999992911e-05
25	8.0100000000002737e-05
26	8.0011999999996267e-05
27	8.0252000000005839e-05
28	8.4228999999993572e-05
29	9.1807999999997102e-05
30	9.2061000000003214e-05
31	9.1850999999997615e-05
32	9.1976999999996534e-05
33	9.2105000000000899e-05
34	9.239999999999249e-05
35	9.2219999999997621e-05
36	9.1359999999998479e-05
37	9.6119999999992182e-05
38	0.00011171700000001117
39	0.00010742200000002367
40	0.00010732700000004591
41	0.00010741300000005616
42	0.00010717600000009764
43	0.00010846700000000098
44	0.0001081520000000028
45	0.00010888000000000889
46	0.00024501499999995957
47	0.0001194150000000116
48	0.00011883800000000999
49	0.00011896599999994262
50	0.00011919600000009911
51	0.00012022600000005657
52	0.00011898299999990591

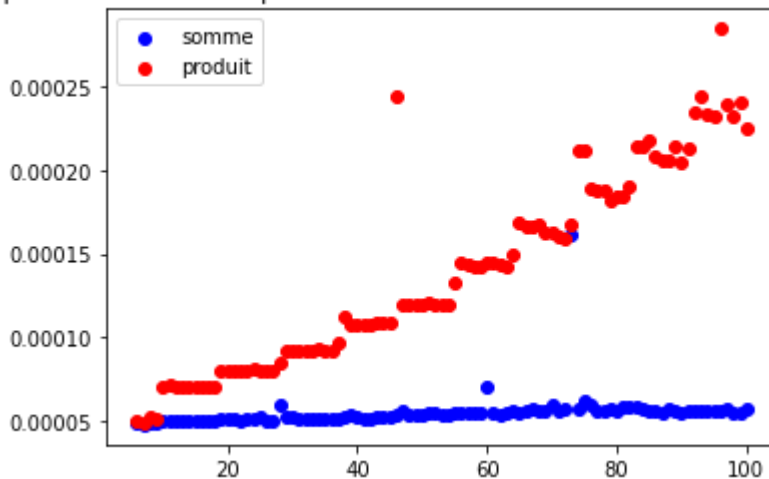
53	0.00011945899999998844
54	0.00011900899999994774
55	0.0001321100000000186
56	0.00014490000000000336
57	0.000143135000000007195
58	0.00014234000000001856
59	0.00014240299999990214
60	0.00014450499999996147
61	0.00014490799999999915
62	0.0001438660000000048
63	0.00014244799999996172
64	0.00014912900000000811
65	0.00016855500000001467
66	0.0001666629999999003
67	0.000166878000000003694
68	0.00016753099999999854
69	0.00016267000000000314
70	0.00016251299999991975
71	0.000159960000000007005
72	0.00015957499999996738
73	0.00016782599999998205
74	0.00021252999999998856
75	0.00021155800000000561
76	0.00018870500000001122
77	0.00018803000000000057
78	0.000188390000000003825
79	0.00018170799999994713
80	0.000184501000000000358
81	0.00018406299999995657
82	0.000190760000000006754
83	0.000214492000000001056
84	0.00021452599999993716
85	0.000217998000000010805
86	0.00020785000000000942
87	0.000205991000000007198
88	0.00020564300000000331
89	0.00021472000000000158
90	0.0002045739999999574
91	0.000212698000000001116
92	0.00023534499999999792
93	0.00024508999999994785
94	0.000233458000000007522
95	0.0002329739999999969
96	0.000285132000000002126
97	0.0002399509999999605
98	0.00023261699999999053
99	0.000240740000000007256
100	0.0002247710000000068

On affiche maintenant les résultats dans deux courbes.

Entrée [5]:

```
import matplotlib.pyplot as plt
x = [n for n in range(6,N+1)]
plt.scatter(x, s, c = 'blue',label = "somme")
plt.scatter(x, p, c = 'red',label = "produit")
plt.legend()
plt.title("Temps d'exéc. somme et produit de 2 entiers en fct de leurs no")
plt.show()
```

Temps d'exéc. somme et produit de 2 entiers en fct de leurs nombres de chiffres



On refait mais en répétant  $10^6$  fois les opérations (sans imprimer les listes, maintenant).

Entrée [6]:

```
N = 100
r = 1000000
s,p = test(N,r)

import matplotlib.pyplot as plt

x = [n for n in range(6,N+1)]
plt.scatter(x, s, c = 'blue',label = "somme")
plt.scatter(x, p, c = 'red',label = "produit")
plt.legend()
plt.title("Temps d'exéc. somme et produit de 2 entiers en fct de leurs no")
plt.show()
```

Temps d'exéc. somme et produit de 2 entiers en fct de leurs nombres de chiffres

