

СПЕЦІАЛЬНІ РОЗДІЛИ

ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №1

Багаторозрядна арифметика

Приходько Юрій ФБ-12

1. Мета роботи

Отримання практичних навичок програмної реалізації багаторозрядної арифметики
Ознайомлення з прийомами ефективної реалізації критичних по часу ділянок програмного коду та методами оцінки їх ефективності.

3. Завдання до комп'ютерного практикуму

А) Згідно варіанту розробити клас чи бібліотеку функцій для роботи з m -бітними цілими числами. Бібліотека повинна підтримувати числа довжини до 2048 біт. Повинні бути реалізовані такі операції:

- переведення малих констант у формат великого числа (зокрема, 0 та 1);
- додавання чисел;
- віднімання чисел;
- множення чисел, піднесення чисел до квадрату;
- ділення чисел, знаходження остачі від ділення;
- піднесення числа до багаторозрядного степеня;
- конвертування (переведення) числа в символьну строку та обернене перетворення символьної строки у число; обов'язкова підтримка шістнадцяткового представлення, бажана – десяткового та двійкового.

Бажано реалізувати такі операції:

- визначення номеру старшого ненульового біта числа
- бітові зсуви (вправо та вліво), які відповідають діленню та множенню на степені двійки.

Б) Проконтролювати коректність реалізації алгоритмів

Продумати та реалізувати свої тести на коректність.

В) Обчислити середній час виконання реалізованих арифметичних операцій. Підрахувати кількість тактів процесора (або інших одиниць виміру часу) на кожну операцію. Результати подати у вигляді таблиць або діаграм.

Хід роботи

Після написання бібліотеки класу `bignum` створимо файл тестування що перевірить коректність нашої роботи. В ньому ми порівнюємо значення обраховані за допомогою бібліотеки та без неї, а також комутативність операцій.

[illegible][illegible][illegible][illegible]

```
In [37]: (bn(3) ** Dbn).base10()
Out[37]: 26588814358957503287787

In [38]: 3**D
Out[38]: 26588814358957503287787

In [39]:
```

Також підрахуємо середній час за певну кількість виконань для кожної з операцій.

Ось результати роботи tests.py що містить тести на правильність

```
[*] Checking the correctness of the conversion...
A == Abn: True
A16 == Abn16: True
A2 == Abn2: True
[!] Conversion to common bases seems right
[*] Checking addition...
Abn + Bbn == Bbn + Abn: True
A + B == Abn + Bbn: True
(A + B) + C == Abn + (Bbn + Cbn): True
D + A == Dbn + Abn, where D is negative number: True
[!] Addition seems right checking subtraction...
Abn - Bbn == Bbn - Abn: True
A - B == Abn - Bbn: True
(A - B) - C == Abn - (Bbn - Cbn): True
D - A == Dbn - Abn, where D is negative number: True
[!] Subtraction seems right
[*] Checking multiplication...
Abn * Bbn == Bbn * Abn: True
A * B == Abn * Bbn: True
(A * B) - C == Abn - (Bbn - Cbn): True
Abn * 123 == Abn+Abn...+Abn times 123: True
(Abn+Bbn)*Cbn == Abn*Cbn + Bbn*Cbn: True
[!] Multiplication seems right
[*] Checking division...
Abn > Bbn: True
Abn // Bbn == A // B: True
Abn < Bbn: False
Bbn // Abn == B // A: True
Abn % Bbn == A % B: True
[!] Division seems right
[*] Checking power...
base**Bbn == base**B: True
[!] Seems right
[*] Starting time tests
Average addition time: 0.000000149577 seconds
Average subtraction time: 0.000000157708 seconds
Average multiplication time: 0.0000003258695 seconds
Average division time: 0.000000504860 seconds
Average powering time: 0.0000006867900 seconds
```

Не впевнений в правильності підрахунку часу за допомогою timer() в пітоні, тому запустимо профайлер.

Для додавання:

```
107 function calls in 0.000 seconds

Ordered by: standard name

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
1      0.000    0.000    0.000    0.000 <string>:1(<module>)
1      0.000    0.000    0.000    0.000 bignum.py:10(__init__)
1      0.000    0.000    0.000    0.000 bignum.py:52(__add__)
1      0.000    0.000    0.000    0.000 conv_types.py:15(convert)
1      0.000    0.000    0.000    0.000 prof_tests.py:9(add)
1      0.000    0.000    0.000    0.000 {built-in method builtins.exec}
2      0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
65     0.000    0.000    0.000    0.000 {built-in method builtins.len}
1      0.000    0.000    0.000    0.000 {built-in method builtins.max}
32     0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
1      0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
```

Для віднімання

207 function calls (206 primitive calls) in 0.000 seconds					
Ordered by: standard name					
ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.000	0.000	<string>:1(<module>)
1	0.000	0.000	0.000	0.000	bignum.py:10(__init__)
2	0.000	0.000	0.000	0.000	bignum.py:77(sub_s)
2/1	0.000	0.000	0.000	0.000	bignum.py:93(__sub__)
1	0.000	0.000	0.000	0.000	conv_types.py:15(convert)
1	0.000	0.000	0.000	0.000	prof_tests.py:11(sub)
1	0.000	0.000	0.000	0.000	{built-in method builtins.exec}
2	0.000	0.000	0.000	0.000	{built-in method builtins.isinstance}
129	0.000	0.000	0.000	0.000	{built-in method builtins.len}
2	0.000	0.000	0.000	0.000	{built-in method builtins.max}
64	0.000	0.000	0.000	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}

Для множення

62354 function calls (60990 primitive calls) in 0.022 seconds					
Ordered by: standard name					
ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.022	0.022	<string>:1(<module>)
6143	0.005	0.000	0.010	0.000	bignum.py:10(__init__)
1024	0.001	0.000	0.001	0.000	bignum.py:147(mulStep)
1	0.000	0.000	0.022	0.022	bignum.py:160(__mul__)
1365/1	0.003	0.000	0.022	0.022	bignum.py:250(karatsubaStep)
1706	0.000	0.000	0.000	0.000	bignum.py:275(shiftLeft)
2047	0.006	0.000	0.011	0.000	bignum.py:52(__add__)
6143	0.003	0.000	0.004	0.000	conv_types.py:15(convert)
1024	0.000	0.000	0.000	0.000	conv_types.py:3(getDigits)
1	0.000	0.000	0.022	0.022	prof_tests.py:13(mul)
1	0.000	0.000	0.022	0.022	{built-in method builtins.exec}
11262	0.001	0.000	0.001	0.000	{built-in method builtins.isinstance}
20159	0.001	0.000	0.001	0.000	{built-in method builtins.len}
2389	0.000	0.000	0.000	0.000	{built-in method builtins.max}
9087	0.001	0.000	0.001	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}

Для ділення

5760826 function calls in 3.384 seconds					
Ordered by: standard name					
ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	3.384	3.384	<string>:1(<module>)
3874	0.004	0.000	0.088	0.000	bignum.py:10(__init__)
1	0.000	0.000	0.000	0.000	bignum.py:134(__lt__)
1929	0.000	0.000	0.000	0.000	bignum.py:141(__eq__)
1	0.008	0.008	3.384	3.384	bignum.py:207(divMod)
1	0.000	0.000	3.384	3.384	bignum.py:234(__truediv__)
1927	0.000	0.000	0.363	0.000	bignum.py:242(lshift)
3855	0.001	0.000	0.961	0.000	bignum.py:246(rshift)
7710	0.562	0.000	0.562	0.000	bignum.py:26(base10)
1927	0.362	0.000	0.363	0.000	bignum.py:281(lshiftBits)
3855	0.959	0.000	0.960	0.000	bignum.py:299(rshiftBits)
971	0.018	0.000	0.026	0.000	bignum.py:52(__add__)
972	1.081	0.001	1.379	0.001	bignum.py:77(sub_s)
971	0.001	0.000	1.461	0.002	bignum.py:93(__sub__)
3874	0.083	0.000	0.083	0.000	conv_types.py:15(convert)
1930	0.000	0.000	0.000	0.000	conv_types.py:3(getDigits)
1	0.000	0.000	3.384	3.384	prof_tests.py:15(div)
1	0.000	0.000	3.384	3.384	{built-in method builtins.exec}
5818	0.001	0.000	0.001	0.000	{built-in method builtins.isinstance}
3816040	0.211	0.000	0.211	0.000	{built-in method builtins.len}
1943	0.000	0.000	0.000	0.000	{built-in method builtins.max}
1903223	0.092	0.000	0.092	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}

Для піднесення в степінь

1340682 function calls (1311462 primitive calls) in 0.444 seconds

Ordered by: standard name

ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.444	0.444	<string>:1(<module>)
131542	0.106	0.000	0.203	0.000	bignum.py:10(__init__)
3	0.000	0.000	0.000	0.000	bignum.py:141(__eq__)
21924	0.014	0.000	0.017	0.000	bignum.py:147(mulStep)
9	0.000	0.000	0.444	0.049	bignum.py:160(__mul__)
1	0.000	0.000	0.444	0.444	bignum.py:181(__pow__)
29229/9	0.061	0.000	0.444	0.049	bignum.py:250(karatSubaStep)
36534	0.007	0.000	0.007	0.000	bignum.py:275(shiftLeft)
1	0.000	0.000	0.000	0.000	bignum.py:32(baseN)
43839	0.123	0.000	0.226	0.000	bignum.py:52(__add__)
131542	0.067	0.000	0.088	0.000	conv_types.py:15(convert)
21929	0.003	0.000	0.003	0.000	conv_types.py:3(getDigits)
1	0.000	0.000	0.444	0.444	prof_tests.py:17(pow)
1	0.000	0.000	0.444	0.444	{built-in method builtins.exec}
241155	0.018	0.000	0.018	0.000	{built-in method builtins.isinstance}
435820	0.025	0.000	0.025	0.000	{built-in method builtins.len}
51153	0.008	0.000	0.008	0.000	{built-in method builtins.max}
1	0.000	0.000	0.000	0.000	{built-in method math.log}
195994	0.012	0.000	0.012	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}
1	0.000	0.000	0.000	0.000	{method 'find' of 'str' objects}
1	0.000	0.000	0.000	0.000	{method 'rstrip' of 'str' objects}