

СПЕЦІАЛЬНІ РОЗДІЛИ ОБЧИСЛЮВАЛЬНОЇ МАТЕМАТИКИ

Комп'ютерний практикум №1

Багаторозрядна арифметика

ФБ-23 Моїсеєнко Дмитро

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.

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:

- 1) переведення малих констант у формат великого числа (зокрема, 0 та 1);
- 2) додавання чисел;
- 3) віднімання чисел;
- 4) множення чисел, піднесення чисел до квадрату;
- 5) ділення чисел, знаходження остачі від ділення;
- 6) піднесення числа до багаторозрядного степеня;
- 7) конвертування (переведення) числа в символну строку та обернене перетворення

символьної строки у число; обов'язкова підтримка шістнадцяткового представлення, бажана –

десятькового та двійкового.

Бажано реалізувати такі операції:

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

Б) Проконтролювати коректність реалізації алгоритмів; наприклад, для декількох багаторозрядних a, b, c, n перевірити тотожності:

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:

Напишемо бібліотеку класу `bigint.py` і створю файл тестування для перевірки коректності цієї роботи. Також порівняємо значення обчислення за допомогою бібліотеки `bigint` і без неї і комутативність операцій.

Результати роботи:

[illegible]

[illegible]


```
dmitry@dmitry-virtual-machine: ~/laba1
In [9]: Abn.base10()
Out[9]: 1170180236371856889837651525949827761218256211117868767875230663903278901656559684806509608749393859016494404551538314515856
733694788034871648056087000084190993998387478477196465586139661615873057703242814680694914419065249466544317561535524400405596552073
636302344090656925538674471218368725287304525181295029150069898732435960592694347027497163292840046505605893422837857431798696984400
867366404781690241953447492694139489236892842320690497726912548782167744104866294201602290546268989379829304847821337839963771589994
6038082219113899251567954657086453434403081597044510585186025684618921192841938

In [10]: A >> 65
Out[10]: 365681323866205278074266101859321175380705065974333989961009582469774656767674901502034252734185580942654501422355723286205
229279621260897390017527187526309685624496087024123895495668644254960330532263379587717160755957890458295099237979851375126748922523
011344482528330289230835772255740226652282664119154696609405280853886237685216983446092863529012514533001841694636830447437092807625
271052001494278200610452341466918590386529013225215780539660171494427420032770716938000715795709059181196657764944168074988678621873
313690069347309351611498583033951669825096299907640955787063302644341287276310

In [11]: Bbn.base10()
Out[11]: 531229615392203808715706807869172465702474806522413794787244139029684496375640812650881833242691513266743512529940846199133
37013428766271426704434231043828031925992081317076965589400047618611424033102071171509022916335066911412954542533178340605435629247
504666769447389446263457688570014309900970785272894109672163750388778170716732283694178904911378181543742465477124393237028626350266
189006034274601629947276727337527357771073615690002232385888482474716397803028263587377133942227995321933177866406407574123842009634
201303596746779281108560229721827313555733281128793446244613063214753064194049606393439452613810

In [12]: B << 65
Out[12]: 195989135190302800218061739726839831824353915165421079696246003319236038192247668974360799144899609830461920824015816760752
076981323773936594772816970287760842832982949511982164474565129872973610375656805297316140940812326215980638295747070493988085047011
944348617713749780539721694433855457627267928658245474516215393041688203713712831909218610578203423048530560815601107181919938430249
151537689041453637294387224935793881082408732504538078553779221039249075837292890839695368345732375994193978008606106226526307787323
6464735455359467324401364773778106723499237851927564501968795565189638948675782258069200352877585741696828618833920

In [13]: (bn(3) ** Dbn).base10()
Out[13]: 1

In [14]: 3**D
Out[14]: 1

[4]+ Stopped ipython3
dmitry@dmitry-virtual-machine:~/laba1$
```

Результати роботи Test.py там містить тести на правильність:

```
dmitry@dmitry-virtual-machine:~/laba1$ python3 Test.py
[*] 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
```

```
[!] 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: 0.000000221915 seconds
Average subtraction time: 0.000000362042 seconds
Average multiplication time: 0.000007993334 seconds
Average division time: 0.000001240298 seconds
Average powering time: 0.000016302200 seconds
```

Тепер результати Test1.1.py

Додавання:

```
dmity@dmity-virtual-machine:~/labai$ python3 Test1.1.py
108 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  Test1.1.py:9(add)
1      0.000    0.000    0.000    0.000  bignum.py:12(__init__)
1      0.000    0.000    0.000    0.000  bignum.py:57(__add__)
1      0.000    0.000    0.000    0.000  convtypes.py:15(convert)
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}
33     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}

108 function calls in 0.000 seconds
```

Віднімання:

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	Test1.1.py:11(sub)
1	0.000	0.000	0.000	0.000	bignum.py:12(__init__)
1	0.000	0.000	0.000	0.000	bignum.py:82(sub_s)
1	0.000	0.000	0.000	0.000	bignum.py:98(__sub__)
1	0.000	0.000	0.000	0.000	convtypes.py:15(convert)
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}

107990 function calls (105644 primitive calls) in 0.098 seconds

Множення:

Ordered by: standard name

ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.097	0.097	<string>:1(<module>)
1	0.000	0.000	0.097	0.097	Test1.1.py:13(mul)
10562	0.020	0.000	0.039	0.000	bignum.py:12(__init__)
783	0.002	0.000	0.002	0.000	bignum.py:158(mulStep)
1174/1	0.005	0.000	0.097	0.097	bignum.py:171(__mul__)
1174/1	0.012	0.000	0.097	0.097	bignum.py:270(karatSubaStep)
2737	0.003	0.000	0.003	0.000	bignum.py:28(base10)
1565	0.001	0.000	0.001	0.000	bignum.py:310(shiftLeft)
3129	0.020	0.000	0.036	0.000	bignum.py:57(__add__)
782	0.005	0.000	0.007	0.000	bignum.py:82(sub_s)
782	0.002	0.000	0.012	0.000	bignum.py:98(__sub__)
10562	0.013	0.000	0.017	0.000	convtypes.py:15(convert)
1174	0.001	0.000	0.001	0.000	convtypes.py:3(getDigits)
1	0.000	0.000	0.098	0.098	{built-in method builtins.exec}
19950	0.003	0.000	0.003	0.000	{built-in method builtins.isinstance}
33866	0.005	0.000	0.005	0.000	{built-in method builtins.len}
5476	0.003	0.000	0.003	0.000	{built-in method builtins.max}
14270	0.002	0.000	0.002	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}

5914317 function calls in 6.775 seconds

Ділення:

5914317 function calls in 6.775 seconds					
Ordered by: standard name					
ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	6.775	6.775	<string>:1(<module>)
1	0.000	0.000	6.775	6.775	Test1.1.py:15(div)
3905	0.026	0.000	0.239	0.000	bignum.py:12(__init__)
1	0.000	0.000	0.000	0.000	bignum.py:139(__lt__)
1924	0.031	0.000	0.048	0.000	bignum.py:146(__eq__)
1924	0.007	0.000	0.007	0.000	bignum.py:147(<listcomp>)
1924	0.001	0.000	0.001	0.000	bignum.py:148(<listcomp>)
1	0.044	0.044	6.775	6.775	bignum.py:225(divMod)
1	0.000	0.000	6.775	6.775	bignum.py:254(__truediv__)
1921	0.002	0.000	0.723	0.000	bignum.py:262(lshift)
3843	0.006	0.000	1.800	0.000	bignum.py:266(rshift)
7686	1.270	0.000	1.270	0.000	bignum.py:28(base10)
1921	0.719	0.000	0.721	0.000	bignum.py:316(lshiftBits)
3843	1.791	0.000	1.794	0.000	bignum.py:334(rshiftBits)
989	0.047	0.000	0.073	0.000	bignum.py:57(__add__)
990	1.943	0.002	2.580	0.003	bignum.py:82(sub_s)
989	0.008	0.000	2.795	0.003	bignum.py:98(__sub__)
3905	0.206	0.000	0.211	0.000	convtypes.py:15(convert)
1925	0.002	0.000	0.002	0.000	convtypes.py:3(getDigits)
1	0.000	0.000	6.775	6.775	{built-in method builtins.exec}
5885	0.003	0.000	0.003	0.000	{built-in method builtins.isinstance}
3906428	0.440	0.000	0.440	0.000	{built-in method builtins.len}
1979	0.006	0.000	0.006	0.000	{built-in method builtins.max}
1932569	0.215	0.000	0.215	0.000	{method 'append' of 'list' objects}
1	0.000	0.000	0.000	0.000	{method 'disable' of '_lsprof.Profiler' objects}
29760	0.005	0.000	0.005	0.000	{method 'pop' of 'list' objects}

Піднесення до степеня:

dmitry@dmitry-virtual-machine: ~/lab1					
29760	0.005	0.000	0.005	0.000	{method 'pop' of 'list' objects}
622101 function calls (608967 primitive calls) in 0.561 seconds					
Ordered by: standard name					
ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	0.561	0.561	<string>:1(<module>)
1	0.000	0.000	0.561	0.561	Test1.1.py:17(pow)
59170	0.117	0.000	0.224	0.000	bignum.py:12(__init__)
3	0.000	0.000	0.000	0.000	bignum.py:146(__eq__)
3	0.000	0.000	0.000	0.000	bignum.py:147(<listcomp>)
3	0.000	0.000	0.000	0.000	bignum.py:148(<listcomp>)
4390	0.011	0.000	0.013	0.000	bignum.py:158(mulStep)
6579/12	0.031	0.000	0.561	0.047	bignum.py:171(__mul__)
1	0.000	0.000	0.561	0.561	bignum.py:199(__pow__)
6579/12	0.067	0.000	0.561	0.047	bignum.py:270(karatSubaStep)
15323	0.018	0.000	0.018	0.000	bignum.py:28(base10)
8768	0.007	0.000	0.007	0.000	bignum.py:310(shiftLeft)
1	0.000	0.000	0.000	0.000	bignum.py:34(baseN)
17524	0.123	0.000	0.226	0.000	bignum.py:57(__add__)
4378	0.026	0.000	0.034	0.000	bignum.py:82(sub_s)
4378	0.008	0.000	0.060	0.000	bignum.py:98(__sub__)
59170	0.073	0.000	0.096	0.000	convtypes.py:15(convert)
6584	0.004	0.000	0.004	0.000	convtypes.py:3(getDigits)
1	0.000	0.000	0.561	0.561	{built-in method builtins.exec}
111756	0.019	0.000	0.019	0.000	{built-in method builtins.isinstance}
204669	0.028	0.000	0.028	0.000	{built-in method builtins.len}
30670	0.016	0.000	0.016	0.000	{built-in method builtins.max}
1	0.000	0.000	0.000	0.000	{built-in method math.log}
82145	0.013	0.000	0.013	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}

