Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

**“Спеціальні розділи обчислювальної математики”**

Комп’ютерний практикум

Робота №2. Багаторозрядна модульна арифметика

Виконала
студентка гр. ФБ-11 Данькова Єлізавета

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.

2. Завдання до комп’ютерного практикуму

А) Доопрацювати бібліотеку для роботи з m-бітними цілими числами, створену на комп’ютерному практикумі No1, додавши до неї такі операції:

1) обчислення НСД та НСК двох чисел;

2) додавання чисел за модулем;

3) віднімання чисел за модулем;

4) множення чисел та піднесення чисел до квадрату за модулем;

5) піднесення числа до багаторозрядного степеня d по модулю n .

Модулярну арифметику рекомендовано реалізовувати на базі редукції Баррета, піднесення до степеня – на базі схеми Горнера.

ХІД РОБОТИ:

**НСД**

```
def GCD(a, b):
    divisor = [1]
    compare = 0
    col_vo_sub = 0
    while a[0] % 2 == 0 and b[0] % 2 == 0:
        a = LongShiftBitsToLow(a, 1)
        b = LongShiftBitsToLow(b, 1)
        divisor = LongShiftBitsToHigh(divisor, 1)
    while a[0] % 2 == 0:
        a = LongShiftBitsToLow(a, 1)
    while LongCompare(b, convert_from_hex('0')) != 0:
        compare += 1
        while b[0] % 2 == 0:
            b = LongShiftBitsToLow(b, 1)
        compare_of_number = LongCompare(a, b)
        compare += 1
        if compare_of_number == 1:
            min_ab = b
            sub = LongSubstration(a, b)
            col_vo_sub += 1
        elif compare_of_number == -1:
            min_ab = a
            sub = LongSubstration(b, a)
            col_vo_sub += 1
        else:
            min_ab = b
            sub = [0]
        a = min_ab
        b = sub
```

```
    divisor = LongMultiply(divisor, a)
    return divisor, compare, col_vo_sub
```

Алгоритм:

1. Ініціалізація змінних:

- `divisor` - ініціалізується значенням 1, яке буде результатом - найбільшим спільним дільником.

- `compare` - відповідає за підрахунок порівнянь чисел під час виконання алгоритму.

- `col_vo_sub` - кількість віднімань під час виконання алгоритму.

2. Перша фаза алгоритму:

- Поки обидва числа `a` і `b` парні (діляться на 2), вони кожен раз зсуваються вправо на один біт, а `divisor` зсувається вліво на один біт. Це дозволяє виокремити загальний чинник 2.

3. Редукція `a` до непарного числа:

- Поки `a` парне, воно зсувається вправо на один біт, щоб зробити його непарним. Це допомагає пропустити зайві операції ділення на 2.

4. Основний цикл алгоритму Евкліда:

- Доти, поки `b` не стане рівним 0, виконується наступне:

  - Порівнюється `a` та `b`, щоб знайти менше з них.

  - Віднімається менше з них від більшого.

  - Результат віднімання стає новим `a`, а менше число стає новим `b`.

5. Результат:

- Коли `b` стає рівним 0, значення `a` є НСД `a` і `b`.

- `divisor` містить НСД `a` і `b`.

- `compare` містить загальну кількість порівнянь чисел під час виконання алгоритму.

- `col_vo_sub` містить кількість віднімань, які були зроблені під час виконання алгоритму.

## НСК

```
def LCM(a, b):
    gcd = GCD(a, b)[0]
```

```
    multiply = LongMultiply(a, b)
    result = LongDivideModule(multiply, gcd)[0]
    return result
```

## Додавання

```
def LongAdititonModule(a, b, mod):
    sum = LongAddition(a, b)
    result = LongDivideModule(sum, mod)[1]
    return result
```

## Віднімання

```
def LongSubstractionModule(a, b, mod):
    if LongCompare(a, b) == -1:
        while LongCompare(a, b) == -1:
            a = LongAddition(a, mod)
        result = LongSubstration(a, b)
    else:
        sub = LongSubstration(a, b)
        result = LongDivideModule(sub, mod)[1]
    return result
```

## Множення

```
def LongMultiplyModule(a, b, mod):
    μ = evaluateMu(mod)
    mul = LongMultiply(a, b)
    mul_mod = BarrettReduction(mul, mod, μ)
    return mul_mod


def LongSquareMod(a, mod):
    sq = LongMultiplyModule(a, a, mod)
    return sq
```

## Степінь

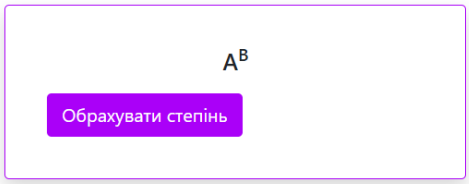
```
def LongModulePower(a, b, mod):
    a_mod = LongDivideModule(a, mod)[1]
    b_mod = LongDivideModule(b, mod)[1]
    pow = [1]
    μ = evaluateMu(mod)
    for i in range(BitLength(b_mod)):
        if BitCheck(b_mod, i) == 1:
            pow = BarrettReduction(LongMultiply(pow, a_mod), mod, μ)
        a_mod = BarrettReduction(LongSquare(a_mod), mod, μ)
    return pow
```

## Редукція Барета

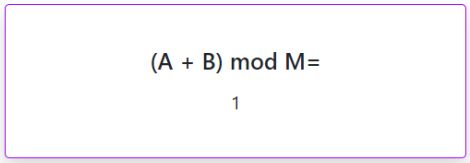
```
def BarrettReduction(value, module, mu):
    k = len(module)
    q = LongShiftBitsToLow(value.copy(), (k - 1) * 32)
    q = LongMultiply(q, mu)
    q = LongShiftBitsToLow(q, (k + 1) * 32)
```

```
    reduction = LongSubstration(value.copy(), LongMultiply(q, module))
    while LongCompare(reduction, module) >= 0:
        reduction = LongSubstration(reduction, module)
    return reduction
```

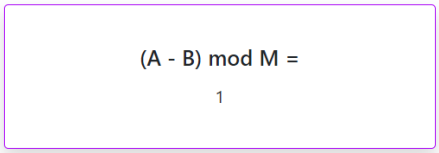
## Приклад виконання:




```
(A+B)modModule = 1
Time taken for (A+B)modModule: 0.12289667129516602 seconds
(A-B)modModule = 1
Time taken for (A-B)modModule: 0.13137292861938477 seconds
```



Performance calculation example

Operand length: 1024 bits

Experiment count: 1000

Using modulo:
13036224528568456303865325802596038899495302773559990284829933312095574721514673982428070276044617603863675414085479868516228064209270936588112122694562171586288721481284523491035103548856398673222662278064238402785798160059069783652701514626455027189631752659352947727692493499941243793729049201148226768553 8

Running 1000 experiments on 1024-bit bigints; operation: +

- Average operation duration: 0.0 ns

Running 1000 experiments on 1024-bit bigints; operation: -

- Average operation duration: 997.0664978027344 ns

Running 1000 experiments on 1024-bit bigints; operation: *

- Average operation duration: 2080.9173583984375 ns

Experiment count: 10000

Using modulo:
12035074403464015095991804468314691466138654224507038326026823637441406305830663959429158570391702232448153168299685870124549155012175987149887969612056382104291296287322847647740795916977303737598165518497234030489757595532519117513898566388519285169875676774342414003719756562242260938347663953953 9548832263953

Running 10000 experiments on 1024-bit bigints; operation: +

- Average operation duration: 413.06018829345703 ns

Running 10000 experiments on 1024-bit bigints; operation: -

- Average operation duration: 393.7721252441406 ns

Running 10000 experiments on 1024-bit bigints; operation: *

- Average operation duration: 2226.4719009399414 ns

Experiment count: 100000

Using modulo:
11852034992637502304386105323788163062738334332600820635885826215662406626519292878416754311261816621601645359319015075939030298206580688304039978235414929952459037306135808493480581582768982297998800214224351915243958045319000020823640548112842522308373292658035395541147926836262562884943419526994180784570 0

Running 100000 experiments on 1024-bit bigints; operation: +

- Average operation duration: 169.3129539489746 ns

Running 100000 experiments on 1024-bit bigints; operation: -

- Average operation duration: 269.1960334777832 ns

Running 100000 experiments on 1024-bit bigints; operation: *

- Average operation duration: 2143.537998199463 ns

Operand length: 2048 bits

Experiment count: 1000

Using modulo:
21258650511549875301214376119454440553041338045568191390498354079833262674528849714247172313128461483201094779795398953653973582601394572417131698558531938572271826371480765752720413218840152460165960903245386847826511693027278884162112522771930038050573157373267017069793241046243000702309659997683253821486979774658764540407699184873725245705032874267120416406880109574990784199184609530141857375861315678970878921268937080909723586524266955257183283628971656870452244172804036407205846240562259808503484880925678333766160762247241715984825274237312963245173220732385030439406548725664474361076203377293999822595926

Running 1000 experiments on 2048-bit bigints; operation: +

- Average operation duration: 1026.153564453125 ns

Running 1000 experiments on 2048-bit bigints; operation: -

- Average operation duration: 999.2122650146484 ns

Running 1000 experiments on 2048-bit bigints; operation: *

- Average operation duration: 12328.386306762695 ns

Experiment count: 10000

Using modulo:
317156982193706370926981305735860805133174188288518845023078337379
816861387908974250863613997771516138054637814965900459933425479965
401585981554444183746096938867650115820899713661192082403997274573
051082246107468789664612212527097288224409257219294581301350412595
504136605100264888046236318256160064202342507337797406939729966592
832878263685683263834512883108193953593039511715008491713591609641
043219268625613988580832588302388100127008194842899681478397436649
558493653728533850328653420730623279525608076698172592515823610973
130363202017881901544358710361083121937161588051379446550242185165
5152381340060459143609 2

Running 10000 experiments on 2048-bit bigints; operation: +

- Average operation duration: 154.47139739990234 ns

Running 10000 experiments on 2048-bit bigints; operation: -

- Average operation duration: 200.0570297241211 ns

Running 10000 experiments on 2048-bit bigints; operation: *

- Average operation duration: 8214.282989501953 ns

Experiment count: 100000

Using modulo:
641853088374148046778824429993750501399185018717338313320788421164
437973806985650558835076222670272909825426434056480491961736521982
473332631003865954369437909054670874131999966102141219444762433486
262727021744892515273286218002084371258345217831929483593755478083
269832447027806171244731086456755717135789899985172792884031517839
987616972456695347241511580909480480682641690936941233405534459679
958288160290988699831991824121587349346336878077755247709044211736
067131424829611401681054514237904071352877718105908089839601986550
493086982685286229487706589113900738772777232745816474920950804703
6151980583265541011794

Running 100000 experiments on 2048-bit bigints; operation: +

- Average operation duration: 254.5762062072754 ns

Running 100000 experiments on 2048-bit bigints; operation: -

- Average operation duration: 378.5562515258789 ns

Running 100000 experiments on 2048-bit bigints; operation: *

- Average operation duration: 7315.189838409424 ns

Operand length: 4096 bits

Experiment count: 1000

Using modulo:
157551475024035927923617363714794387907959259611272879233601866459
009824580026893462337379456251589336974051759510498761162246525107
927490942603322688171894923513660963830648976632936244892952612124
241276322784138712001771819427522445682279267199422158202233152662
880674951708486949012666320171336069783796558801759745034573381093
654316392746700693369494215697091592697329851361461281431683682729
409498076362522550258100419651174751796221275937845255813074445844
966155231286715104406209713469887144592685278265505510845798351679
466681401535859652029883189503077092491943491170042347839758788773
077244465188794047296958358808098336424513834730729120434919378546
273224810434401570482368721956204130509350670953197035243672397748
489701803151762674441061933440965467819506338009359976519873935486
339298931005332061626678787681216029489667285629804207145185930179
905619977776022285115369366833424471924960346945781214999710887326
240334436213764192178713027995225716425675077452706641755007707842
862856769359738865553070339804929156644159547407302527423018822728
371575780048403494017172948258209077453659332753588541873465483182
977641731431632160689006243866601338859953508354161002951987075290
909796783396852254243727466326047423297736376

Running 1000 experiments on 4096-bit bigints; operation: +

- Average operation duration: 995.3975677490234 ns

Running 1000 experiments on 4096-bit bigints; operation: -

- Average operation duration: 0.0 ns

Running 1000 experiments on 4096-bit bigints; operation: *

- Average operation duration: 20390.748977661133 ns

Experiment count: 10000

Using modulo:
84378773631803024634132376438395375893291705517712501125840397350290335394528020852007555743086679999207795487666065575006106864971427653590276702768029650636360888821477927170335359408118786255978646990439716113851706148944371477674903248076961294205519094812491929026208738479019903001835302810114395754107643004520437532910557613097333694055719063397363522616276325305199280405736533935305887803159205043876148536325721874536906905639728637911763934358363750070649769633014386815326055433395319916570708696744504828258108883027144153555076780314254880408406496660802711917817435520056163847569901729838610205366050991847641032608737313386273673616295475514631794294113908922304532050184704217803161194947362648592871214784035344606693218992503324440451053605505169653214718211537944707784369580458508723167508685304595831775589960081751583140997729887267164774363587460027562946729871854517999068688774605481113533812397917258760237835610756665910882979307330259275622954573491136621392966936360640759517261783372175846699365613668252255517815123598688188148749056282100861612160044610910668886237666898388083082723537390766476794122703726653658031010602441527046548605215334901474990151449775320008574075270729577355224176551676 5

Running 10000 experiments on 4096-bit bigints; operation: +

- Average operation duration: 301.361083984375 ns

Running 10000 experiments on 4096-bit bigints; operation: -

- Average operation duration: 504.4221878051758 ns

Running 10000 experiments on 4096-bit bigints; operation: *

- Average operation duration: 20652.50873565674 ns

Experiment count: 100000

Using modulo:
34945250606951966429775817206327235185373858145597757348160928395859303668292373858436793164007363604237900444890362600885973093910906754410653518762841915493298717394634557443340426105414848510903254414745234147262246255124488768481555224450884618548110637676577528888239298307142447329446179762406677773936230701650756112278139596380516340704796403341352378523984399683298872754173120052728307633576564318549073897252953822176022203969624690887662902262686297 5

75629316481999941866293493833323087887143979948134238772501653467363659345019282175142177096839460983306447364236609035737428568635215878077006035252033500748812672257572947054067377936250678948668983522698204648691819625780654066382063038669365864371978458257317700515822264081031777901553006958191991322711231976741212653233629652987884985178407542897776670755375054620615992128863528833969426165645282322808408303510261026493152789217136524706959131239349162382563727851616251987515136052697911337610295289819123547817348400398307218079242528797502073720478575505888491169607416684590557738443314152873118576863724504780387484600733578659536653568686677087881997799403355171489935084348284307068413600613970514640706936015278906799869979607219250480645201087674699628 2

Running 100000 experiments on 4096-bit bigints; operation: +

- Average operation duration: 475.9860038757324 ns

Running 100000 experiments on 4096-bit bigints; operation: -

- Average operation duration: 621.0446357727051 ns

Running 100000 experiments on 4096-bit bigints; operation: *

- Average operation duration: 21164.85834121704 ns