Московский Авиационный Институт
(национальный исследовательский университет)

Институт информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу “Криптография”

Студент: Жерлыгин М.А.
Преподаватель: Борисов А.В.
Группа: М8О-308Б-18
Дата:_____
Оценка:_____
Подпись:_____

Вариант №6.

Задача:
Разложить каждое из чисел представленные ниже на нетривиальные сомножители.

Первое число:

123248268911937923199906141216645363665087045422689358104089185316148911496103

Второе число:

158968690785896053229304195025980740908911607577490592481192836972930851627507291449244730388234361901478286114627747425663442922357381267982988585772251423678977375807360238275429639874676052862046713568690409185767729868661335316050142125453936421554346233052917382325478595789259674397146933105369462870471989751163449408072638444931191132643054360803184618121059080807310404316851562692251939368391798187363382805306816975035313741234210109232681400128607993 1

Описание

Факторизация - процесс разложение числа на его простые сомножители. Для решение поставленной задачи существует множество алгоритмов.

Например, для первого числа можно применить алгоритм p - Полларда, как один из наиболее простых и эффективных. Но этот алгоритм эффективен для чисел, порядок которых меньше порядка числа из задания.

Тогда я узнал про библиотеку **msieve**, реализованную на Си, которая использует метод квадратичного решета и ряд других алгоритмов для разложения чисел с большим количеством знаков. С помощью msieve мне удалось разложить первое число за 2 мин. 47 сек.

Однако эффективно эта библиотека может раскладывать числа порядка не более 110 (как указано в документации), а второе число имеет порядок 400 знаков. Его факторизация на обычном компьютере практически не реализуема ни одним существующим алгоритмом. С помощью подсказки можно понять, что один из множителей второго числа определяется как

наибольший общий делитель с одним из чисел другого варианта. Поэтому можно написать программу, перебирающую все числа другого варианта и определяющую НОД и числом моего варианта. Она выводит его, если он не равен единице. Второй множитель получается как результат деления изначального числа на НОД.

Тестирование

mmaxim2710@DESKTOP-RDPBU3D:/mnt/c/Users/mmaxi/Desktop/coursera/msieve-master$ ./msieve -v 123248268911937923199906141216645363665087045422689358104089185316148911496103

Msieve v. 1.46
Thu Mar 4 20:35:39 2021
random seeds: 7a55615d c4af38b6
factoring 123248268911937923199906141216645363665087045422689358104089185316148911496103 (78 digits)
no P-1/P+1/ECM available, skipping
commencing quadratic sieve (78-digit input)
using multiplier of 1
using 32kb Intel Core sieve core
sieve interval: 12 blocks of size 32768
processing polynomials in batches of 17
using a sieve bound of 958673 (37824 primes)
using large prime bound of 95867300 (26 bits)
using trial factoring cutoff of 27 bits
polynomial 'A' values have 10 factors

sieving in progress (press Ctrl-C to pause)
38153 relations (19419 full + 18734 combined from 208196 partial), need 37920
38153 relations (19419 full + 18734 combined from 208196 partial), need 37920
sieving complete, commencing postprocessing
begin with 227615 relations
reduce to 54555 relations in 2 passes
attempting to read 54555 relations

recovered 54555 relations
recovered 43377 polynomials
attempting to build 38153 cycles
found 38153 cycles in 1 passes
distribution of cycle lengths:
   length 1 : 19419
   length 2 : 18734
largest cycle: 2 relations
matrix is 37824 x 38153 (5.6 MB) with weight 1155520 (30.29/col)
sparse part has weight 1155520 (30.29/col)
filtering completed in 3 passes
matrix is 26990 x 27052 (4.3 MB) with weight 912181 (33.72/col)
sparse part has weight 912181 (33.72/col)
saving the first 48 matrix rows for later
matrix is 26942 x 27052 (2.9 MB) with weight 677958 (25.06/col)
sparse part has weight 490464 (18.13/col)
matrix includes 64 packed rows
commencing Lanczos iteration
memory use: 4.1 MB
lanczos halted after 428 iterations (dim = 26942)
recovered 18 nontrivial dependencies
prp39 factor: 321985376278994307302664413499387768503
prp39 factor: 382775982984847122295865568872934509201
elapsed time 00:02:47

Исходный код

```
from math import gcd


list = []
list.append(2849949678058592728534773278622454669783469198065854321335567699592693152711111)
list.append(3523581180791504931870993551416295271017491061679972555096190205283337223522217)
list.append(1197606395839410537256528037313284196976497391762438410219156212428076186085 91)
list.append(3448452281301592264881635710704176792350251390158020191525169262027118466601 41)
list.append(1607693578999756108281995391141095181675311345141909907851446669320766147178 41)
list.append(2741148223395896290240264954415574797138132280289801178690522789506812411948 19)
```

```
list.append(10876235329244848744124766368551365889316764693062717894612888996764317215412 7)
list.append(26888732002909002811721449825320409576588413648336619384236128377650064396678 1)
list.append(12324826891193792319990614121664536366508704542268935810408918531614891149610 3)
list.append(28499496780585927285347732786224546697834691980658543213355676995926931527111 1)
list.append(47237955273687149405814323916262286089696527511354345058027248989166708020776 3)
list.append(36199672745678487185560418105660567208862266620757816081129106087399715170888 7)
list.append(31323089459651394116306551650054215948186184975398206471670692604095575391260 1)
list.append(37445690250873943521827325867122445734134840648853318819552882781962751323326 9)
list.append(61121970174911146319545193754425119520875945215282784640177276523929376501913)
list.append(38345661488490246672625273129454423465801539061937283582624662549915438411818 9)
list.append(24258741345568931180594169758210354434344402573793060972812930301130760182355 1)
list.append(18155287756599894391061854322552857993532144720973697891248911845081854523048 9)
list.append(31937361327089666376595411565492262487935984166599285265812448737288112357000 3)
list.append(37445690250873943521827325867122445734134840648853318819552882781962751323326 9)
list.append(16951284854020837637732470255086077812968838518009345966053244779029899896723900984413142336870385225437965243629326745116590849908770944614057690683052539801654819522761512642822701693074249824513493644688844526263633663327921066974983001545042891090435383147221714908515772020029364695158378468844726857013205559546752704709817118834528761529676361607229919430317377276744622348039645465223497066788134123417127031908420255679798222788292548376427537395466491 59)
list.append(19162420871806801568617129945097280525351590911288448056586790252967165594044346648117256191866527259013257746490175941447883606374071784769363169152207581445356819643713116570717509704147072181122222804539518752135916397350198445796426220148742125948380414578004649211823451274964608882500841718155403512117458135421929696241085675044819052903173594157525350779859315079097221673643129800998340230230212127671070403013443927834175759810 02
```

59379669074442689507301)
list.append(15987565442108608120026832525046666312840385351549793409109648246739235786392263979181344291927370058541881779770591778582438559908039812756656909129755340910413617018434655781017338634797816807916559595783204421083716340483743135242021931986948945364524716468688251447430144529579127439202399544735343744226477480201653067693793961900445995131103930624613028392443567547410653207750115147747231558637315951828928227907098432963750752726519026414605041032917753 61)
list.append(12501714973722279820265559996751701089479189513783673434709234831041585972166320665863009215668112657764654273950264581512400423660612715121077525866816999239149020618862130225444967830707270610837639966308162798691691946231692557111354225219254441359390148782775152998705368759629482679738995456217285477265451923825939369855749788813059494875232331486771063306508182234439558006227741899366351063630357846982161854615737617147662116078126952812523566744324442 79)
list.append(15987565442108608120026832525046666312840385351549793409109648246739235786392263979181344291927370058541881779770591778582438559908039812756656909129755340910413617018434655781017338634797816807916559595783204421083716340483743135242021931986948945364524716468688251447430144529579127439202399544735343744226477480201653067693793961900445995131103930624613028392443567547410653207750115147747231558637315951828928227907098432963750752726519026414605041032917753 61)
list.append(16117655691488048562428673842586807198500102862981912046351541529420432197290447526886147483136114545465725205417369977940016871273001825655775233013745768986374654630793295442477747872835121549831617371165626457442345657277097463641140055832315479670230254145694131224473280404169708453094322175307224333415061668790581352676527375610862399155982339310065668240742080964683365204046938632685331174477299911625792360364160144090922283544048098857799988000765501 37)
list.append(14177467869787507650387834432016948376930580071471350079285831921442569467042236590494758980427157782351530260852126352560893481056955596585856196760851613464821804136259107185547729368883111388512812700339059708262004996928275687558408584407339919174540282553261747449656964703936447130918315087871163722894672660845644433050799802860493503622897613938633077951879747971879859575334614760888258163959225587279203300668232112105942963026762617074322173483051121 87)
list.append(15896869078589605322930419502598074090891160757749059248119283697293085162750729144924473038823436190147828611462774742

56634429223573812679829885857722514236789773758073602382754296398746760528620467135686904091857677298686613353160501421254539364215543462330529173823254785957892596743971469331053694628704719897511634494080726384449311911326430543608031846181210590808073104043168515626922519393683917981873633828053068169750353137412342101092326814001286079931)
list.append(1447056357743040318789862961227509104744799081494678612383291986984923519316446287708049077918224656527429543673229364351887183390807262752423117298211041934655152276599225431751671588895981517419026215429324481989444969083616331327076407980393565709505006078950141506587407820420736302617335256351925247739018311504537066619041864399051765841946047321403468580781936233573521469460165494767804910732129539946607701693482114451990193860694698453061853232064399 61)
list.append(1262485504020168731000842257581537957328326497522478405002465359648875356810280292244547618070727524417592419776792612058732594852983180148665064058817407866064291179552422627557683886828462061069447032164569235069818669414169882863307032697282802157247652797734392044016320040859257401114524063142894607111829574025600918893253339517061607978068475589931239014683019592991614837523358909806258991077646147246997493894736434495372693444001308001278879395788963879)
list.append(1916242087180680156861712994509728052535159091128844805658679025296716559404434664811725619186652725901325774649017594144788360637407178476936316915220758144535681964371311657071750970414707218112222280453951875213591639735019844579642622014874212594838041457800464921182345127496460888250084171815540351211745813542192969624108567504481905290317359415752535077985931507909722167364312980099834023023021212767107040301344392783417575981002593796696074442689507301)
list.append(1960344000673448010109966123798259138788312223000110285444138984687043682091918437726564873652655959337927213942829283843615252926281789196372471730892422452230531118265385923148587364956392045025267762404119597838874471039017253236308306374541274375355671500991196394524509192278487473429022067848460150114918996838415401644820324493941862061208584686840594025223786924079444262714095490301772077126395790235999836003971290616988894725373002042174148527448991721)
list.append(1688432268535652536976161544225404933352917348466880741646555236080940468369390533777566901374863846088926302716704958253349013465017171687476514345454080829512228091554390695242226222710223271367480753308157792549868681240943730184545304781633011043927327584175094195702062946904306735673354996415907014195550 5)

90550472255345961637249641012928019098518193363458415691802243705768503786649882426739768062694678022813527067727278842446759998639312587246098493677573)
list.append(16698120282111148760357415934740218022123400447408847013442712701958320858567973149367256099699198928804324700476844645415672653368067889584026253505220722153568875423450919653644127144161477230078248529404392103475354920799309387153018516635049076332711782159866874962816730597954315002008001123373748886576429320113770107797396319904117148857373617146027153976389826461364816302384194818088644389113718040852129468401985584414791762568326896004766689308652227 09)
list.append(14169084447719341143272360643356951750338555687245147232760909092389022494507611631161792983700976377366095987469785396811908061750237394378249497902031141954472876211921620528639113700302812533115824770238590279848186791082392676007634118911135781819389783413687636778555346854134274372902392765730783654373168911955055844636426697161129367283730885533859028643592189337506274405214704767741787934130977543281068768100090834326282132886721944207546209205488511 29)
list.append(15109385843025147460686876803591387120848268695317498338161525361070299566943782286650144848099932846806364650453365846700065126924820571688588052517305224124355755370476387591838494378611695821743531006167608614420833389111629829780186546090734874556183447256464743411064487701861194654374368055403145739023151480106056429693990362392799908664813775526310383450383326713004604491508261330475994029527022043813232424080148048305599685013560938061277308857626493 9)
list.append(15033499906313505127942896843130782450400802347937492882843881028115293186513341863145092476540091725800064574393561521328602108813560462716993292012530584330361623216743051882118851116282237496594977166868163840331786137975686189271737528006927952316222354349343355004965993153577865952088162134894290906187247294161317469653368470808158015990205733111105113749510972776073107995920975786223684234407181645957200918994270361355390957408076391671959950085809104 33)
list.append(15406225094908179490535246491659819823627101385901456801083676605923009421074555721288749853133175961437905691459584340477262214693347263610455103335077100041993455290528251101324297893844389907069275490156858433293934015201542373720790986683248170112968253028251028561435745802110146265300808951070303379347718785548915972867011694369046393627681841310401546991286457558418299989014726795657302590989466425402873370832842632874326160946972589939452327670137815 01)

```
list.append(16265705923840344012310598594084552548100509114311455807738173203854456785977766950683127961452586180126554485218163161080222787625202392679798991846278167936565809063790777458251309334207819198020137034051556960335295557939983593891737558873668573291313432061486325062585463987617255877140830088283477270714347719449643648297679057781869127718984315010760253784801108318403324790207832062061904051003949822187692691563935316046036041428410910392654850704146722591)
list.append(13421244726926808148646960398316572013419301705374908889481859091934049269612235364794740406664608853763787428191971057486507988658794823047710842362566543842737997023364759696474517001465344300528458335539841650829928420258996565679227744843136567637933476457628883796920941366450593437711550436902196628304015729314682900588143047244398242042599808167107962408352746044620761234616984773844437137512844829946074307551558342332836812531322809893949894719618171431)


if __name__ == '__main__':
    with open('test2', 'r') as file:
        n = int(file.read())


    for i in range(len(list)):
        number = list[i]
        result = gcd(n, number)
        if (result != 1):
            print('1 number:', result)
            print('2 number:', n / result)
            break
```

Ответ:

Разложение первого числа

prp39 factor: 321985376278994307302664413499387768503
prp39 factor: 382775982984847122295865568872934509201

Разложение второго числа

1 number:
130473868032583609827185448980364758181025721979158584058510968480674675318007582512991438379409900025634206602999335047253958595

764575192257231951973654902109838773403174196838685087469564811
369549756562791721156060671611860553422489104596835427639417952 3
648789184332036017207346493555432935800481061311666 49

2 number: 1.2183948646804611e+154

Вывод

В ходе выполнения работы я столкнулся с сложностью факторизации настолько больших чисел. Второе число получилось факторизовать только прибегнув к хиторсти.