Relatório

Aluno: Renan Rocha Souto dos Santos

1. Números aleatórios

Os algoritmos escolhidos para este trabalho foram *Linear congruential generator (LCG)* e *Xorshift*. Os testes foram baseados na geração de 10000000 números pseudo-aleatórios a partir dos *seeds*.

Algoritmo	Tamanho	Tempo (Segundos)
LCG	40	2.859052896499634
LCG	56	2.8702728748321533
LCG	80	2.8896777629852295
LCG	128	2.9372620582580566
LCG	168	2.990936517715454
LCG	224	3.092780590057373
LCG	256	3.154209613800049
LCG	512	3.777458429336548
LCG	1024	4.609506368637085
LCG	2048	6.296304702758789
LCG	4096	10.464219808578491
Xorshift	40	3.3244495391845703
Xorshift	56	3.4553730487823486
Xorshift	80	3.469250440597534
Xorshift	128	3.49381422996521
Xorshift	168	3.6291611194610596
Xorshift	224	3.750485897064209
Xorshift	256	4.084242820739746

Xorshift	512	4.948448181152344
Xorshift	1024	6.3025267124176025
Xorshift	2048	9.233561992645264
Xorshift	4096	15.83188247680664

Para gerar um número a partir do número anterior da sequência baseado em um seed no algoritmo *Linear congruential* são necessárias três operações básicas: multiplicação, soma e módulo. Já no algoritmo de *Xorshift* são necessárias sete operações básicas: três de *e* lógico, duas de deslocamento de bits para a esquerda, uma de deslocamento de bits para a direita e módulo.

Contudo, comparando os dois algoritmos, o *Linear congruential generator* apresentou melhores resultados.

2. Números primos

O segundo método escolhido foi o de Fermat. A escolha foi através de pesquisas que apresentaram como um algoritmo básico e bem conhecido. Além disso, o algoritmo apresenta um conteúdo já estudado na disciplina de matemática discreta e, portanto, foi um estímulo para aplicar na prática o que foi visto na teoria.

O algoritmo de Fermat apresenta problemas em alguns casos. Isso acontece pois este algoritmo é baseado em uma sequência de tentativas e, se essas tentativas não cobrir um caso específico, o teste pode falhar.

Foram gerados dez números para cada comprimento de bits. Foi utilizado o algoritmo *Linear congruential generator* para obter os números pseudo-aleatórios e foi verificada a primalidade deles através do algoritmo de Fermat com validação de 1000 (k = 1000). Contudo, o comprimento dos números gerados podem possuir o comprimento informado + 1.

Tempo total de execução para o arquivo de teste: 4396.088068008423 segundos.

Tempo médio para 4096 (+1) bits (geração e validação): 6 minutos.

Saída do algoritmo de teste:

40 bits

1099561848949

1429012753669

1847585772511

1277207860729

1308562861009

1997233185001

1222438768141

1882554796933

1809396726301

1835283399343

----

56 bits

72057598105843069

76878410125884391

130872525532193479

89989061861948461

93014341312920343

116553131437003741

133803118988462479

103184929911414919

120267042004346521

131876717354551339

----

80 bits

1827619865188681627288993

2091236881058928884785633

1265153577526396878946321

1958689827101971110165379

2249292022462025806377613

2331039935432483770315609

1915268922786241848094363

1864262461741809477048619

1472741516354104076353843

2348218081485871753083001

----

128 bits

340282366920938463463374607431818432629

340282370926902920471305839532592148949

414459331713801034932082269533485677529

563465876108119792409126962646245576081

481185387963482714093180480785724301619

659857314803806802095885885451409155413

383351651845085062995074025536919341533

483496209867853316342332626960717148843

357971296424414756211999064536986814613

497073695548130905999249575329740666159

----

168 bits

374144419156711147060143317175382636974954297418789

566696905834304617019243938246245771235589880381841

408835370049864450910644551970583995510714550083823

384213164537683906989715526314955423675418014943019

678138787954492344318951160877454252504021595942501

437120744648192440842927980352808250573930228534611

486248206368117337106741490732454704367225070593843

711205772113466990640795782980224786699103068389191

574875752793611932650970470134958848775233468362921

598125118365889296607482307853750313691339977672793

----

224 bits

46334082482640273834564915912353393963627833730407078200815939185639

42849871210532563895928877221758367022688235859175402359183409764609

28088369485732484783392686779270910082225402031220759102860116452811

47974283784534341479080687215135462215375109029527118666661287437589

30526691138977737233409027089062379619964897430294137668488120537941

47987364134313412021931709558085557562413768951179008966815712877353

31568800573198622790422739476765597904654546005434234531162589354119

48518774101313348105724001370315693727842899986062644826103638451091

52656185890092638059007234964424545132814896311518127983823004374053

38233376378023002780719197188380784339900590306638339945195848347971

----

256 bits

15174102038816984982169744882132250471591598677450547222739242193678
2310080913

13679404820389739219232745232583770943936153207608212206580075251157
6688808283

21275956351752609392211563642507003890462482846940627115786385150071
4926716213

19616336559706352488757370941521566173571902115303603511527177796918
7128276849

13939988071014420404095985887793966886438766503682276726884238705072
3288870513

13330811325330198155433095535262819154534333019110972139609035214573
1287771733

16455314147374390733302172673810669568474688459239531541264299535225
8710310141
21859582356688714141777191842164282857218618177257643781089498088814
3333820659

22135795962190230651396346408841899141517294857594513279856572819633
5882193051

22938660701053472231451097630896634043061094034187082301525450292932
4325216051

----
512 bits

13407807929942597099574031979373589935004801764500907734098661040300
10167468147438776729704087154826186410549430803954810984580457715781
7489537983257723509

19736650313977989560634715859185158329462529686442599364908564776240
42640992239555608623904818177845866561469472248606544468608251557014
3354008961827482609

24852217902888937781892996020726635279822457779761085738796558119566
62115413928521526168217054247949667186400427766888002730028260000943
7208480016257620433

1956605221127885901992459687663930241039974683088046104606113694017899911215863722183231977631524059836844476588821572582566832550435358875223677301098 3441

1612018758179431681149472506355605640258540047528740028983575616023751763106846220438885806141161631438846628461679262819677471140071145952791838330779 7389

1718306958750839379803320870054616704124876563693805391475907224539271738185576372211353224386123131143428341385190594075148325120638080603709007391229 0181

2235650056360170983627450978497229193677015911399391275310406101859853390330817475533913166442151021140965693700614192286250715412248480320554052999123 1711

2120978472427761061584135586476180326582793695592860656806131694731492853257007435351883290247856662428963493239139806887994926308696457629188732806043 6331

1923681863892534664114969417092714294849343565433841447207246261403925496716904588356765532614264136137592651287937173334744442760910040416310432586461 5699

1977279994142524802227220484871541552114335207493466038872376906586087191263333593276083774965327282818588283193720993810539900326106572309782307133087 8799

1024 bits

2691724480606949274998088568273353941675598249691481884063860112025535490118662871519475810351915052006733887909118845612126439415470053729454772531678786112998408473246573500945547576575400376432497216260967851043824295345516168920323556225091277704572155048653780214558412919412071692764994760997052082 64949

2648145354926931540145239649717193185018289097208411170306518392110669367319462289686293719666942448332043246907818116120175978110126305767215062485385627161232844125013813515246435025553388848333874483407238644345646443664169637641082931746875596743013158983854867482233361691163881248479341483331060438 45379

2136449151565383603544637831753747138478635430091734835908550934989341521064930286874368882428848384942492805259571288094612801531916418073792857031677047763954772176115484633321450885568569966056479289 76

593170472283675676651279974014930026437719474844513862372919012673006761930537137255633387876562136688533

199628376533914850693686093819268669107617187600202092902128323280832704762514475994897974384729907456240268331515830401961391142346006086256564752648079060271799763192232237402466044536237822650670994347632827498771677919097582467550283435713230832282125353661919471598383297274457508719203731964556244861063

309772768813479696685268065345862744009160623085410188894280186841552900028302900167254773788985113120661114622059463907350853130040533984997287624813178006419500703255147305451597868202200709581396168906618743839334473679399414394722510679954681095035205673007698354626051393027367618675477715026483388606611

258428241536755971557149773245641954541819950187250455912748466446838500560979861394306241008500020252609577206012277298324595348317999968035700092079568322242509321837482378494930889064596072003313556782138226500445466391864192330791762494021921273239389186145798250861422569940098746973700418573235916898531

341985554062459294555372230040269464880520236156991880159049935218310671386148062701746818466828355180265255640477871282174995072381277132971618113009363354946323663204214401000210359621463902441479679401750017103122373019524367085241778516340789427995481190433067509964725677873131468982943690041575640221621

336243057067204186255231218014415631725900890122710492020258724630464678187648717179029053999165410426852742906954924373253533488873377079917728990779826766391351870257365297462415624757451152060168279065067622239159515565852615973121311728758762936417236499670654502813057600734299713090514070778520052193903

283929737167859940181896559308598896962204351667230673481996604256606125537184964937201207501825101830451497184837728877321330071832692087449923008365402858322685727549452992710908594934920534783508551308445098855737468864920391356596149142452095070422722618665093695655470275107951468638502447102881011749089

191757727506712989290180296366553393114096275540552721413474192087848491218554900862308739215148818735545181168402958221351702543407339619758046068310362270111037355190457267051882964017259601593276544153194510848962891754665821886585672774574894913142179036783246831054493703748223986676413118772086415192301

2048 bits

40784261232585532397751967154592652473484007712883913151181826731662
22359046816878282107225774201181794048294221007477238558949089004909
76952765575796882030560436413877718600676396609405150044462595464290
58620942041128827014821663100779089061660427287391298553184483032500
29227277346156905619839731299223673560924979497453675532635127130326
49613842072972085379663340394209053775569508141128508991301337535890
11880577013576750078720712155860637932875867260663148816873643548724
70980554688662853490437653243208993376310312823856309782996717968947
95209250886230465482259078399373398494834938064868548065962579565312
98891

45175661591966079501730465490644400930172136475976830083822157169343
50600328802224384347247311726799792350397971812681288851412644578205
14669876036335216241458761255399049887915576626195690792201742537606
02887931784068250911677055528556327171827315406845337415928721127569
53444114272062008125285567488324758506055724547981714617702479648041
97906071101099111809411404215368223118040585799723307698387797445629
60382724389194279066704946547914890137062219779980188246349343813679
38998491128611678026890722037976170901132730909922969968480020955986
32128931587815096358798768946770100395098815135863566276440009015013
3929

42024032327510869768961342455598116869845034862055642311658160693210
92226634141783399299615674976777397872117761034327469949688817822786
50154269015162988924690061215716748111769269765468852689168533878585
15371620302154063071230380468327045375690958577979899971547128861773
00178606483278616248748019959908831307754055539838477490677183797128
94413688737226136029666257816353470344772178551792811232725392832519
02950123088618776157349514965787244716145874160316832159154430787266
46820400273121433162607526070811927684159818729944956446837077445517
19110091975388326248594778938396284439388912432725924053923649626500
78711

48445244611576960501906456769820464942056281844274191274398149991876
98721038740175235606474618040311181016828821033571313446396161726417
26052034146083001518046676983682380935744735550758683706021942820872
37737543082773870976909462935726944109338825586241299309675652547793
52123021748712695085781915539439346001477060672299136412153525314734
72983931385086035649771054643120193434111148092197398190968724161276
76456184179122776527316064823952358191295269749926431412339046314271
46321685242371096412553556323695441730401387772789760138336058769009
02289641177820541811466615551812814290946660738020285181656347828409
39919

5146107608275628161342951564137558117059958387092291958143302133588725938717807266032937599019601987742698134491136075789050012309111370998628245762579437198763532376320855582524692097922906605917850332805873357130390915261758548710849422529536782287922428748113375999898253680936413703084438775185574716439111745513238690176195395544847367490446470598256929053306003607015627159388538747634099163473185428988912437564133586957147265685625356630008702807184867867701249069612123293392952532046816529471568782060452236780747919471713166740090638074782257570284108572317134835932258223842581211408183892587595405641865

6278505731431001237124873939471592689279831513557352573001903969803906748647451873724117044029374238886077512481873985976173992235708327306820008168986658738126844529223638555942789891576461026179614510164190144796687505734169943787170609250514921969091398416704494138410647600814255018811078249301708485023304046337721228121656612880301074576089652458415247629102683210994569532894101725750821101490484785124343095154697933956667126126536220947470057609869639236984436697798879261358832335650190818302465648141351122047334794483518371303489459152180066699857515719894138864944529327225415712533129084016803400278963

4329743558494020698855141023497042293360338517840153879326875770526377844036639627193671258540314085560612823877130062105719581440279998101640115108783916612291262899224305464086793663174043620680122378955154177063072260461698268740474023053079700903874895671284037719419300168390497746279450308677414781362138303119059879188571484902101709309913460600108581420131561592007283994546913338358058306310798873931125168726377385940048096348610974203758009757789372303425290037149930125115320459182099597603694362807561479826213638090696969118598783756739225149648859302665128468746274500682592815389447032479468483559234

4158922240624434151221101497277632196899774353032888051295010111196276989758846503799924828425184374559671307821662839966908116806417684782210508637812824264580208809074044252126708456686274444960761713117966295653176892068168686440869898664937649001491088000276410793354108570515677615990311258581253082465580968074010415054134438744701209365757052888896599893427545082990684986862320819340670377061764389937818299555416338901155362267054645194714359087649111216507855565983970201790965280387609083111245057375469116060874969158867602653619299820218331230197193056351994431822977523484750527246013631104626919275

8403

5798947888784617296672377761511318812469933022933507170754695998718022014811721689612064010073725447845126984771834364250684409951013855

4524658572955556044447627272229468512758276535243487933696935456198803316466733740773361890980885137199373391363600877990464926872791017112886070588418735033901627101017035295706501736929209200962371031616534024259413267789934043006078181252904685325764501792057024230640308447281953460013186167629689875113582634220849579792550034807764550537953500012444281399753736399426627259854751699596073692660952274635176317087112247170714032687828425318034397403798094199995221198411215509

4795285289768279388452535908245953908963309967250285153719904500823239774348984325091184236650007091568499623337967592841841768717838394450517239071306309535950534936744492696596165262190731733877363948268934166041803236490866271885697180555167339196683023826176046419724707137719330179910030006626112313690680557358078465006482111065206311263667735931587127011292564633385320631043311888235254092621711505525861553160314289421292412709858951083379855213182869452368640640639997178280610306859536338265927746770792412550886662127906903528647024417837965785919897783804675505657088278001807488799778536799777978213 3559

4096 bits

1044388881413152506691752710716624382579964249047383780384233483283953907971557456848826811934997558340890106714439262837987573438185793607263236087851365277945956976543709998340361590134383718314428070011855946226376318839397712745672334684344586617496807908705803704071284048740118609114467977783598029006686938976881787785946905630190260940599579453432823469303026696443059025015972399867714215541693835559885291486318237914434496734087811872639496475100189041349008417061675093668333850551032972088269550769983616369411933015213796825837188091833656751221318492846368125550225998300412344784862595674492194617023806505913245610825731835380087608622102834270197698202313169017678006675195485079921636419370285375124784014907159135459982790513399611551794271106831134090584272884279791554849782954323534517065223269061394905987693002122963395687782878948440616007412945674919823050571642377154816321380631049134385173973826985294077652977183090488191327426626363310649028248583075098200096117533662303819370167731107739012532785884546706543088494224556640812723251822742479208701269712884225782628911871676375136316579546469369922863707373871257091817665521255404807023913101127988042962406911498520594220441530604455117047654424869

1712676870269811027454582214659260395084711182853027102377607563723904247036851093661603659641751211098606334232023642856216977720923213253789890419694429163489967486217392640873869136976416731337389564 29

51577027541005825679659686220098277918855449025584584269735123411346
06027107972695528620604506988905146559138988022235241959352295323905
22660923550872683253636603987197083615278700201116549231496019850214
69477527217567912732576396737873680243082430972116030333343946543503
47773432885804715998628088418201122846544639258195988441257829168425
33861057038537430924112216305574346136807198493802931626138169958131
56257965449408428034056804930822979133461569176020744110584045613596
50776588154667286696915871706990769491936493923758165174305307582180
91882748471055716943119731569175845552078395896673914086410385171036
48550772130524613149515809130588447392553340024080930153064508233764
61537435624775010458085075713554780814215525671639148533990181751181
51865358324035564604042369429456780928881679877204026603223839081228
70713216791403800102667911942012350648937230370397975901910977753882
87241371775050084267931840713853800205238384670994824522048749377187
83978139188344978701054654966786705124397886507981283057503393877317
1258167783

12207557443669471747454863978164332576328395206530921807405675696538
58491603103146373000238157826712879865860600032952282080379595509517
87123716183028641250743121876236117804301918637136379389008113830357
16420530478233552203092396269993526400969215478597970625101164481634
35222798564837189652882675854892977011252638282737515313138154502275
58969260810874382240181113675435649016673522170036486467610024582094
30074104804595230081180707597439622369198690573111329903973834592774
76204898136312885366564355248417766358725887589488907621334385918741
08119631120589019076018870837862606888384703398723846206919071060474
72433754789331951096471630808506986552260634178147873290734590215686
66924083476140117942772625776427442272052937722306809735942555472930
94473290102872596996897102023927620825096414680693734438189723978513
87467384031335915829289128861568409353783403503852058473488177467050
84456294637377900190785622204612792665010634037923899274585892350292
67989306253251187548012367506742485525503031115020389688271458180734
38690277708223857451181334425248261564755463496573140941170391331109
38221212883338572094698769872875980515212474965969249811866517133748
16365025693165983643482801843690094621117939909943201879319597501906
2752720371

17451149063159916517468592806987606927049505357253573403830561470080
05404594935731992248023079906300058379609135681442876008203725906202
72038371279009201678931880067129240075787118178398183287884147664491
03104522120002212714061188262058282344602715019613937731726287105674
79171823133829640135000952646025807062129053784588326901809506946740
87398961379207694023189465668045563225956950810495313886011669260600
19267104022952792982495751876995234664767432154351238626739200615754
33409680938360993632364049449420123278246745856013111583504891842140
89841664075075391085459699904317032544634172516353595291797947289389

08129187147275026393190455091593903933506113429046287200386802773289
47502619008655094166919673458675101649815004267737464740428160189360
41905496687128687486819707474473305010099861425190356331326333177856
59320391936181838289930880299090338561956488414550930619950727706502
96648986060657473145274948988415085997842246978054215006095520898484
86190339455523608743177223280079773943236258534575205410975055887487
38645814650068771521105405057687800920488436851465369231892068163318
06587583931304470452590203657241764154012914705770273030562037634772
99574955586345990270991520545991393933899125621356509644339080863887
9766081719

17495223885526552749783019864856582042216261767947275352917988273998
84706415453102661480783612724824587471663316921868531832691483322049
88869447127122630589760392058770274899208091870770045215110919520148
63951916271142742056222927088923060027392842487513001888361297126393
98345006705677421491075443468590841815341197852740795056052339716769
41113530714429627072098258544092989548361981990412303654824210396312
88086669836481771742990761039965670653244759573161957887401262082878
71798566988826336282861046576842054691920266419167602568837041115569
76585713151554358453135079508165581518383315631139238485394480328499
68439374387419844316556409144835793817983259993647056646339080086500
13646706765052956013989575075144358153750300091088176466102797520866
20668356656370913973671501662232790622693716732784832145052519585441
43834156773923007289333108907952380751527311110860214253182922020107
85865341322447857947464990092793826818258441103030737199015728227946
47839829021106971370709616912798478305430706379889442012902369458901
59690718720303699993408666277584776756272452726843685592296739968883
00067364813255630329293708378801544571377906478618927081962456214611
74043533153200051904912654517756889219873665162546482525631224471937
8060456949

20208816960654531303090613154353692536962669696568306854846947620287
50321969454693300696778864968393352536202845488976543310208628752311
08272896625471025051461145792241459153560631998091614637665985490731
13911367918794859975833267185663955159839885056693450613415303914933
39211306891378263649942645178991661820926594876419546772308080401197
55264939959469610556556232070676924900301104963450440425605588216801
82949795252600119973849343374752074808533999914721578610241713674323
50677521621292725759377701650706989575603710956123054451859733072037
78115253050606499379189955325279520925426541538704332216757333239381
52066591709405029658389578626574372420587478983372766347109878000921
73595824895023608559583658647980610428446603287958558504235268408483
18023992470745901767416761182550641455627323031912682102533345207237
20815679836169203053470803162218443464546569291923485118192191689641
44811151400527272924017034962180714403996500100344311241032244059903
18570458707346770702358289683904946755346110303445330370465668306851

52119657037328643218621500979527664496221171170722452681474867377376
07203486189381548169144953766397612265011325684766480737872007695314
95345578087897053903325995675211833260365986720510182329235719712096
4230865083

13954845172408477383287883341738246816319456365223112939888403914804
36080098929700613449851941350967214872444594861757086621773367366187
48360422346378706161939925943631582675122415878460563382919146861650
65555370320650860442858915529533426419310436976815650198488959183331
48042435928178395704319181667967216129688216267020271689711050048405
56557371069571746565671114556835714676883941329964758355322016496566
95893075532419296713631140550452874564241995559933089445056118403209
63207997727127469093362366904094383165932875675004561130854796653121
125718116729615017599459546118973251634867676871186264332311524281824
00290062495194679471573095626299492930263952190151308948715334489411
20366197952369628150389306034081786055267571074705598401809349570684
45312615319294414064254993179804915716985775518702913302317326928793
89620304875663635152786833362688536917837953909209460928601650499669
16422365840269979242651642676323283545619846601655526311284391806488
35872938871368945648158475619064238586285221472204246086031479412249
10915661336663200911821143531504530752885561678618559139358132554548
80885617325309488328952939713632597235015874628758435024632417985415
90961641206824114725190212042535218901835902351809085588209837391107
504985693

11254842620050580624979246760480393774230048252571201936408802469135
21972510298047959596454139894916352139774614074127924280591566194216
77215810904104344700119335248034427275556549132062574360829261353512
06066928520055328915990891019299729076016994935253093970548793010034
11438731508079901037348416813623108526214677144250576809256456813671
80584436511088396032097440349170109901391169104503433933779151633729
20772439526527771408441364558997151862380164714862785198366347498434
44269240628833929473993137367999637450865968830575714976633557430622
85707087127724979973621729665653716288514839188557725168338442674331
388727058009411192766431109724840001465356284268049040726019223864711
09696876869972756984385345018839337564254968028363823003892853974522
56576869572665682938668719556905410360452415940550408938454949476973
46531173692794635873973401952466255892289955737508852931371851314642
99514354575107181645312138024905378849759098578006652370655460311270
27175621608142497106824174085958909121678986000967760568363323439621
73026030483794773172154944352009715228825431103331597092477556791263
13163236034031732994627468543427044958222130439249051535991074476860
90833186142348380120916778764086335232553541513195806892993438091470
859774551

13820612661464610480272135370603812219890822297697983789217791456538

84483451303622175616496049738401623087459439400065188891478262405646
99928793497996058930222620868837782496741678714263627193741396798052
76807682448414019701771424835473275402608750943641059207913676895544
02422698318787053416456566877773042730398858714087089343349130757629
70519315957301588456630640693549061519910386679939524372549238047152
23811966284675012830011739745675136721635861745700175154441954680723
24098182653064391134036019004572543813882346971656236193906077382468
13810592764850915793034339295142613719150364861096824867796196601237
89138505179154397302813352197048657293148889290871658376855735425151
27225985440481038698801525159410779026933236666063176412915871382331
10652461886544242986081901627873221868246929072585091111215330848069
62449183791092325209104988247135958116781878886814564712733638036075
35532851038929732942955094260782892919578525613430252442110396808833
74147754500440241340371735476477743875457023121457930236473216037121
79482959788883653079014395788868133624118580503545743151916934598470
88762795848808305554685774741785131880968761695321715371743968011405
94147467476413854478508441699410592673790458466975052009022186833720
9849584341

13621705214098069676275441521306860986503694344545378868007752641160
45415231756071597555095037520038749582889037817852260592139442444510
42418305682278662712920553171258054708421765476444455258523892819968
20027346972574755688108212597172651910310331386336725172480698195583
25889456258811428016171915951786993594820612414122023444306058483972
10195352821572951925643172608244456439159621854157537626888248242532
39687934895784437848561580429146001159988482203186109019941453164279
21761970280933039128482324229180661486647533553127761893675784530889
97593583826668010733201574865593921652807563043490005007752320269914
60345177072912233994423819421714680417168425927177250476538816601712
60923911314917715876622605148463793849050655258336645238038285361778
69763217969057965523142135317717367929747736762734086167102677584782
91559404797686167651301303403035307830845407303284533888695890427272
29330471834471091170397548059725692495871492679443105922448583070966
95168081677849557013355231640152605222939356504704372533404559127821
05129365698168590223421919827893275357827498070459141239145952791019
59918292003301376800309582734385029569243072282106543441502199340597
07626066294590108304083658567477343822442367913824431092740714702077
6913512531

3. Códigos

Os algoritmos para geração de números pseudo-aleatórios usaram como *seed* o valor 3. Eles são baseados na criação de geradores em Python através da palavra-chave *yield*, em que facilita a implementação sem precisar ter que armazenar vários números que não são utilizados e permite o uso de *Multithreading* de maneira facilitada, pois bastam as *threads* utilizarem o mesmo objeto gerador e invocar o método *yield* para gerar um número. Para os algoritmos de teste de primalidade foi necessário obter um número gerado aleatoriamente para a verificação de acordo com os algoritmos, porém para não utilizar a biblioteca de números aleatórios do Python, foi utilizado como base o *timestamp* para ter algum número que possa representar alguma aleatoriedade. Além disso, foi necessário utilizar um método denominado *power* para realizar exponenciação modulares de números grandes, pois utilizando as formas tradicionais geram *overflow* (apresentado nas referências). Também foi utilizado o algoritmo de Euclides para o cálculo de MDC no algoritmo de Fermat.

3.1 Linear congruential generator

```
def linear_congruential(m, a, c, previous):
        return ((previous * a) + c) % m

def generator_linear_congruential(length):

    # Seed value
    x0 = 3

    # Modulus parameter
    m = 2 ** (length) - 1

    # Multiplier term
    a = 3

    # Increment term
    c = 1

    # Initializing with x0
    previous = x0

    while True:
        # add m for a min of: 2^(n) - 1
        previous = m + linear_congruential(m, a, c, previous)
        yield previous


from time import time


if __name__ == '__main__':
    qtt = 10000000
```

```
    lengths = [40, 56, 80, 128, 168, 224, 256, 512, 1024, 2048, 4096]

    for n_bits in lengths:
        start_time = time()
        generator = generator_linear_congruential(n_bits)

        for j in range(qtt):
            next(generator)

        print(f"{n_bits} bits: {time() - start_time} seconds")
```

3.2 Xorshift 32 bits

```
def xorshift32(m, previous):
    previous ^= previous << 13
    previous ^= previous >> 17
    previous ^= previous << 5
    return  previous % m

def generator_xorshift(length):
    # Seed value
    x0 = 3

    # Modulus parameter
    m = 2 ** (length) - 1

    # Initializing with x0
    previous = x0

    while True:
        previous = m + xorshift32(m, previous)
        yield previous


from time import time

if __name__ == '__main__':
    qtt = 10000000

    lengths = [40, 56, 80, 128, 168, 224, 256, 512, 1024, 2048, 4096]

    for n_bits in lengths:
        start_time = time()
```

```
    generator = generator_xorshift(n_bits)

    for j in range(qtt):
        next(generator)

    print(f"{n_bits} bits: {time() - start_time} seconds")
```

3.3 Miller-Rabin

```
from time import time

def get_an_a(n):
    # Get an "a" based on timestamp to use as a "random" number

    timestamp = time() # Example 1639508236.2790089

    # Remove dot
    # Example 16395082362790089
    str_timestamp = str(timestamp)
    index = str_timestamp.index(".")
    timestamp = int(str_timestamp[:index] + str_timestamp[index+1:])

    # a: 1 < a < n - 1
    return 1 + timestamp % (n - 2)

# Modular exponentiation
# to work with larger numbers
def power(x, y, p):

    # Initialize result
    res = 1;

    # Update x if it is more than or
    # equal to p
    x = x % p;
    while (y > 0):

        # If y is odd, multiply
        # x with result
        if (y & 1):
            res = (res * x) % p;

        # y must be even now
```

```
        y = y>>1; # y = y/2
        x = (x * x) % p;

    return res;

def miller_rabin(n):

    # Step 1
    # n - 1 = 2^k * m

    n_1 = n - 1
    k = 0

    while n_1 % (2 ** (k + 1)) == 0:
        k += 1

    m = int(n_1 / (2 ** k))

    # Step 2
    # Pick an a: 1 < a < n - 1

    a = get_an_a(n)

    # Step 3
    # b = a ^ m mod n
    # Probably prime: b = -1, b == n - 1
    # Composite: b = 1

    b = power(a, m, n)

    if b != 1 and b != n_1:

        # Step 4

        # Keep squaring x while one
        # of the following does not happen

        while k < n_1 and b != 1 and b != n_1:
            b = (b * b) % n
            k *= 2

    return b == -1 or b == n_1
```

3.4 Fermat

```
from time import time

# Euclides
def gcd(a, b):
    if a == 0 :
        return b

    return gcd(b % a, a)

def get_an_a(n):
    # Get an "a" based on timestamp to use as a "random" number

    timestamp = time() # Example 1639508236.2790089

    # Remove dot
    # Example 16395082362790089
    str_timestamp = str(timestamp)
    index = str_timestamp.index(".")
    timestamp = int(str_timestamp[:index] + str_timestamp[index+1:])

    # a: 1 < a < n - 1
    return 1 + timestamp % (n - 2)

# Modular exponentiation
# to work with larger numbers
def power(x, y, p):

    # Initialize result
    res = 1;

    # Update x if it is more than or
    # equal to p
    x = x % p;
    while (y > 0):

        # If y is odd, multiply
        # x with result
        if (y & 1):
            res = (res * x) % p;

        # y must be even now
        y = y>>1; # y = y/2
        x = (x * x) % p;

    return res;
```

```
def fermat(n):

    # k: times
    k = 1000

    for i in range(k):
        # Step 1
        # Pick an a: 1 < a < n - 1

        a = get_an_a(n)

        # Step 2
        # Check gcd != 1

        if gcd(a, n) != 1:
            return False

        # Step 3
        # Check a ^ (n - 1) = 1 mod (n)

        if power(a, n - 1, n) != 1:
            return False

    return True
```

3.5 Teste

Os números gerados e o tempo de execução são mantidos em um arquivo chamado de *output.txt.* Este teste pressupõe que o algoritmo de Fermat tenha sido salvo em *fermat.py* e o algoritmo de *Linear Congruential* em *linear_congruential.py.*

```
from fermat import *
from linear_congruential import *

from time import time

qtt_numbers = 10

lengths = [40, 56, 80, 128, 168, 224, 256, 512, 1024, 2048, 4096]

file = open("output.txt", 'w+')

start_time = time()
```

```
for n_bits in lengths:
    print("Initializing:", n_bits)
    content = f"{n_bits} bits \n\n"

    # Create a generator with n bits
    generator = generator_linear_congruential(n_bits)

    # Generate qtt_numbers for each length
    for j in range(qtt_numbers):
        print("Range:", j)

        # Generate a number
        number = next(generator)

        # Repeat until a prime number has not been found
        while number % 2 == 0 or fermat(number) == False:

            # Generate another number
            number = next(generator)

        content += f"{number}\n"

    content += "\n" + "-" * 50 + "\n"
    file.write(content)

file.write(f"\nExecution time: {start_time - time()}")
file.close()
```

4. Referências

Linear Congruence: https://www.geeksforgeeks.org/linear-congruence-method-for-generating-pseudo-random-numbers/

Xorshift: https://en.wikipedia.org/wiki/Xorshift

Miller–Rabin**:** https://www.geeksforgeeks.org/primality-test-set-3-miller-rabin/

Fermat Method: https://www.geeksforgeeks.org/primality-test-set-2-fermet-method/

Euclidean algorithms (Basic and Extended):
https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/

Modular Exponentiation:
https://www.geeksforgeeks.org/modular-exponentiation-power-in-modular-arithmetic/?ref=lbp