アルゴリズム論特講 （塩田）

2006 年 6 月 22 日の課題

課題

(1) 以下のファイルをダウンロードせよ：

- 関数定義部 crypto060622.py
- Pohlig-Hellman 法のサンプルプログラム ph.py

(2) ph.py のパラメータ（幾つまでの素数を小さいと考えるか、法 p のビット数）を変えて実行し、実行時間等について考察せよ。

提出期限：7 月 6 日 （ 512 号室ポストまで ）

Pohlig-Hellman 法のサンプルプログラム

```
#!/bin/env python
#
# ph.py
# Pohlig-Hellman 法

from sys import *
from math import *
from random import *
from time import *
from crypto060622 import *

##### いくつまでの素数を「小さい」と考えるか #####

print
bound = input(' 幾つまでの素数を小さいと考えますか （ 100 以上 ） : ')
if bound < 100:
    exit(0)

##### p - 1 =（小さな素数の積）となる素数 p を作為的に生成 #####

k = input(' 素数 p のビット数を指定してください （ 10 以上 ） : ')
if k < 10:
    exit(0)
print
print ' 素数生成中 ...'
p = 1
while solovaystrassen(p) == 0 or bitlen(p) > 1.05*k:
    a = 1
    while bitlen(a) < k:
        a *= (rand(bound) + 1)
    p = a + 1
print ' 素数 p =',p,' (',bitlen(p),'bits )'

##### p-1 の素因数分解リストを作成 #####

q = []
m = p-1
```

```
r = 2
s = longsqrt(m)
while r <= s:
    if m % r == 0:            # m が r で割り切れれば
        e = 0
        while m % r == 0:
            m /= r
            e += 1
        q.append([r,e])        # 素因子 r と指数 e を登録
        s = longsqrt(m)
    r += 1
if m > 1:
    q.append([m,1])            # 商が残っていればそれも素因子
print
print 'p-1 の素因数分解 :',q

##### 法 p の原始根 g を検索 #####

print
print '原始根計算中 ...'
g = 1                      # 原始根の候補
ok = 0
k = len(q)-1
h = [0] * k
while ok == 0:
    g += 1
    while jacobi(g,p) == 1:        # 平方剰余は原始根にならない
        g += 1
    gg = g
    ok = 1
    f = []
    for re in q[1:]:
        f.append((p-1)/re[0])    # f は (p-1)/r 達のリスト（2 は不要）
    gf = [1] * k
    while f != h:             # 反復２乗法を同時に行う
        for i in range(0,k):
            if f[i] % 2 == 1:
                gf[i] = modmul(gf[i],gg,p)
            f[i] = f[i] / 2
        gg = modmul(gg,gg,p)
    # この時点で gf は g^((p-1)/r) 達のリスト
    for i in range(0,k):
        if gf[i] == 1:
            ok = 0
            break
print '法 p の原始根 g =',g

##### サンプルデータの作成 #####

y = rand(p-1)+1
print
print 'y =',y
print
print '離散対数 x = log_g(y) を求める'

##### Pohlig-Hellman 法 #####

# p-1 の各素因数べき r^e に対して x mod r^e を求めてゆく

print
print 'Pohlig-Hellman 法開始'
```

```
t0 = time()         # 開始時刻
ls = []             # x mod r^e を格納するリスト
for re in q:
    r = re[0]
    e = re[1]
    m = p-1
    mm = m/(r**e)
    ymm = powermod(y,mm,p)
    gmm = powermod(g,mm,p)
    gg = powermod(g,m/r,p)
    xr = 0
    for i in range(0,e):
        m /= r
        yy = moddiv(powermod(ymm,m/mm,p),powermod(gmm,(m/mm)*xr,p),p)
        # yy = moddiv(powermod(y,m,p),powermod(g,m*xr,p),p)
        ggj = 1
        for j in range(0,r):
            if yy == ggj:             # yy と gg^j が一致すれば
                xr += j * (r**i)    # x mod r^(i+1) が確定
                break
            ggj = modmul(ggj,gg,p)
    ls.append([xr,r**e])
    print 'x mod %d^%d = %d' % (r,e,xr)

x = multichinese(ls)
print
print '解     x   =',x
print '検算 : g^x =',powermod(g,x,p)
print '       y   =',y
print '計算時間   =',time()-t0

##### 単純検索 #####

# g^0,g^1,g^2,... の順に y が得られるまで検索する

print
print '単純検索開始'
t0 = time()         # 開始時刻
z = 1
x = 0
while z != y:
    x += 1
    z = modmul(z,g,p)
print '解     x   =',x
print '検算 : g^x =',powermod(g,x,p)
print '       y   =',y
print '計算時間 =',time()-t0
```

実行例

```
##### Pohlig-Hellman 法実行例 #####

幾つまでの素数を小さいと考えますか ( 100 以上 ) : 10000
素数 p のビット数を指定してください ( 10 以上 ) : 25

素数生成中 ...
素数 p = 64243681  ( 26 bits )

p-1 の素因数分解 : [[2, 5], [3, 1], [5, 1], [17, 1], [7873L, 1]]
```

```
原始根計算中 ...
法 p の原始根 g = 7

y = 36454640

離散対数 x = log_g(y) を求める

Pohlig-Hellman 法開始
x mod 2^5 = 4
x mod 3^1 = 2
x mod 5^1 = 2
x mod 17^1 = 13
x mod 7873^1 = 2374

解      x   = 6033092
検算 : g^x = 36454640
       y   = 36454640
計算時間   = 0.0179998874664

単純検索開始
解      x   = 6033092
検算 : g^x = 36454640
       y   = 36454640
計算時間 = 32.0110001564

--------------------------------------------------------------------------------

##### Pohlig-Hellman 法実行例 #####

幾つまでの素数を小さいと考えますか ( 100 以上 ) : 10000000
素数 p のビット数を指定してください ( 10 以上 ) : 1000

素数生成中 ...
素数 p = 19760231768429748842094811216950118816145718330390015467459705080324913
95660678587383664508099362197009688972881524068916189330315063627995288167427910
82588964830206750074398625527715611355078485219138290981345729931727549923593037
23370000486908423017692518817002240857368878752399360000000000000000000000001
( 1015 bits )

p-1 の素因数分解 : [[2, 38], [3, 17], [5, 23], [7, 6], [11, 6], [13, 6], [17, 2],
[23, 2], [29, 1], [31, 3], [37, 1], [47, 2], [53, 1], [61, 1], [67, 2], [71, 1],
[83, 1], [89, 1], [97, 1], [103, 1], [109, 1], [113, 1], [127, 1], [197, 2],
[233, 1], [269, 1], [293, 1], [349, 1], [373, 1], [419, 1], [443, 1], [577, 1],
[709, 1], [953, 1], [2069, 1], [2221, 1], [2861, 1], [3517, 1], [4003, 1], [4759, 1],
[4799, 1], [6197, 1], [6653, 1], [7867, 1], [8387, 1], [8447, 1], [13103,1],
[14747, 1], [16943, 1], [18899, 1], [23669, 1], [25237, 1], [33113, 1], [33751, 1],
[53731, 1], [68687, 1], [123289, 1], [173149, 1], [178567, 1], [191561,1],
[278353, 1], [322433, 1], [382363, 1], [438601, 1], [648107, 1], [660769, 1],
[708473, 1], [1335233, 1], [2640857, 1], [3213101, 1], [9670207, 1], [9951257L, 1]]

原始根計算中 ...
法 p の原始根 g = 82

y = 43031543871860535667658457008891635655853007477094787326648291504006242371625
97835379554919895255229578720464866619272600995076065197967418711843203537467257
75616523023079841532728165395529919110416072765178050013766265754620576217124997
```

875177786358053949230759442530194147351062148612322331373196844524 33

離散対数 x = log_g(y) を求める

Pohlig-Hellman 法開始
x mod 2^38 = 142586713512
x mod 3^17 = 17564096
x mod 5^23 = 11607461964575587
x mod 7^6 = 19306
x mod 11^6 = 1047249
x mod 13^6 = 54786
x mod 17^2 = 25
x mod 23^2 = 488
x mod 29^1 = 24
x mod 31^3 = 29608
x mod 37^1 = 20
x mod 47^2 = 851
x mod 53^1 = 27
x mod 61^1 = 43
x mod 67^2 = 835
x mod 71^1 = 57
x mod 83^1 = 40
x mod 89^1 = 30
x mod 97^1 = 87
x mod 103^1 = 66
x mod 109^1 = 73
x mod 113^1 = 2
x mod 127^1 = 109
x mod 197^2 = 19948
x mod 233^1 = 224
x mod 269^1 = 21
x mod 293^1 = 86
x mod 349^1 = 297
x mod 373^1 = 218
x mod 419^1 = 92
x mod 443^1 = 101
x mod 577^1 = 401
x mod 709^1 = 346
x mod 953^1 = 867
x mod 2069^1 = 491
x mod 2221^1 = 714
x mod 2861^1 = 2120
x mod 3517^1 = 2817
x mod 4003^1 = 2033
x mod 4759^1 = 3426
x mod 4799^1 = 1246
x mod 6197^1 = 5648
x mod 6653^1 = 5315
x mod 7867^1 = 1971
x mod 8387^1 = 2854
x mod 8447^1 = 5444
x mod 13103^1 = 989
x mod 14747^1 = 7791
x mod 16943^1 = 2141
x mod 18899^1 = 1758
x mod 23669^1 = 13350

```
x mod 25237^1 = 1099
x mod 33113^1 = 10201
x mod 33751^1 = 28176
x mod 53731^1 = 14409
x mod 68687^1 = 22224
x mod 123289^1 = 118066
x mod 173149^1 = 155741
x mod 178567^1 = 39725
x mod 191561^1 = 144060
x mod 278353^1 = 112687
x mod 322433^1 = 24500
x mod 382363^1 = 369052
x mod 438601^1 = 356733
x mod 648107^1 = 536216
x mod 660769^1 = 230679
x mod 708473^1 = 295964
x mod 1335233^1 = 839979
x mod 2640857^1 = 653311
x mod 3213101^1 = 1983059
x mod 9670207^1 = 690160
x mod 9951257^1 = 8143335
```

解　　x　= 13020458948415308231442985230085757990256366564098220446311849475467601391186900403578109721590195471216779556532930728616652651472445029636918167703196273983788981287310845382726866937119584828823744483335481059136449672917022848793799264745410774765629504708439987693941043974244174976417986994365402871 2
検算 : g^x = 43031543871860535667658457008891635655853007477094787326648291504006242371625978353795549198952552295787204648666192726009950760651979674187118432035374672575616523023079841532728165395529919110416072765178050013766265754620576217124997875177786358053949230759442530194147351062148612322331373196844524 33
　　y　= 43031543871860535667658457008891635655853007477094787326648291504006242371625978353795549198952552295787204648666192726009950760651979674187118432035374672575616523023079841532728165395529919110416072765178050013766265754620576217124997875177786358053949230759442530194147351062148612322331373196844524 33
計算時間　= 2164.65200019

単純検索開始（たぶん宇宙が終わるまで答えは出ない）