アルゴリズム論特講 （塩田）

2006年5月11日の課題

課題

- 関数定義部 crypto060511.py と雛形プログラム hina0605011.py をダウンロードせよ。
- hina0605011.py において、法演算での逆数を（ユークリッドのアルゴリズムを引用して）計算する関数を完成せよ。
- 法のビット数を取り替えて実行し、単純検索との計算量の違いを実感せよ。

提出期限：未定 （ 512号室ポストまで ）

雛形

```
#!/bin/env python
#
# hina060511.py

from sys import *
from math import *
from random import *
from crypto060511 import *    # crypto060511.py を引用

# mod m での a の逆数を返す関数
# a と m が互いに素でなければエラー出力して exit
def modinv(a,m):
    ls = euclid(a,m)
    if ls[0] != 1:
        print '法と互いに素ではありません。'
        exit(0)
    else:
        ##### ここを埋めよ #####

while 1:
    print
    print '-----'
    k=input('法 m のビット数を入力してください （ 1 以下で終了 ）: ')
    if k<2:
        exit(0)
    m=randbit(k)
    print '法 m =',m
    a = 0
    while(gcd(a,m) != 1):
        a=rand(m)

    print
    print 'ユークリッドのアルゴリズムを用いた逆数計算 :'
```

```
    print 'a =',a
    x=modinv(a,m)
    print 'x = 1/a =',x
    print ' 検算 : ax =',modmul(a,x,m),' mod',m

    print
    print ' 単純検索による逆数計算 :'
    print 'a =',a
    x=1
    while modmul(a,x,m)!=1:
        x=x+1
    print 'x = 1/a =',x
    print ' 検算 : ax =',modmul(a,x,m),' mod',m
```

関数定義部

```
# crypto060511.py

from sys import *
from math import *
from random import *

# a と b の最大公約数を返す関数
def gcd(a,b):
    while b != 0:
        r = a%b
        a = b
        b = r
    return abs(a)

# ユークリッドのアルゴリズム
# d = gcd(a,b) = ax+by となる d,x,y を求め、
# 3つの要素を持つ配列 [d,x,y] を返り値にする
def euclid(a,b):
    if b == 0:
        if a >= 0:
            return [a,1,0]
        else:
            return [-a,-1,0]
    else:
        r0 = a
        r1 = b
        x0 = 1
        x1 = 0
        y0 = 0
        y1 = 1
        while r1 != 0:
            q = r0/r1
            r2 = r0 - q*r1
            x2 = x0 - q*x1
            y2 = y0 - q*y1
            r0 = r1
            r1 = r2
            x0 = x1
            x1 = x2
            y0 = y1
            y1 = y2
        if r0 < 0:
```

```
            r0 = -r0
            x0 = -x0
            y0 = -y0
        return [r0,x0,y0]

# a を法 m の数に取り直す
def mod(a,m):
    return a%m

# 法 m での a+b
def modadd(a,b,m):
    return mod(a+b,m)

# 法 m での a-b
def modsub(a,b,m):
    return mod(a-b,m)

# 法 m での a*b
def modmul(a,b,m):
    return mod(a*b,m)

# 素朴な素数判定法（n が小さい場合）
# n が素数なら 1, そうでなければ 0 を返す
def primetest(n):
    if n <= 1:
        return 0
    if n == 2:
        return 1
    if n % 2 == 0:
        return 0
    h = 1
    m = sqrt(n)
    i = 3L
    while i <= m and h == 1:
        if n % i == 0:
            h = 0
        else:
            i = i + 2
    if h == 1:
        return 1
    else:
        return 0

#  自然数 n 未満の乱数ロング整数を返す関数
def rand(n):
    n0 = n
    x = 0L
    while n0 > 0:
        n0 = n0/10
        x = 10L * x + randint(0,9)    # randint(0,9) は 0 から 9 まで
の乱数整数
    return mod(x,n)

# k ビットの乱数ロング整数を返す関数
def randbit(k):
    x = 1L
    while k > 1:
        k = k - 1
        x = 2L * x + randint(0,1)
    return x
```