アルゴリズム論特講 （塩田）

2006 年 7 月 6 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 crypto060706.py
- mod p^e での平方根を計算するプログラム modsqrt_pe.py

(2) 素数 p のビット数とべき指数 e を取りかえて modsqrt_pe.py を実行し、反復 2 乗法を応用した計算方法と単純検索の計算量の違いを実感せよ。

提出期限：7 月 20 日 （ 512 号室ポストまで ）

実行例

```
素数 p のビット数を指定してください ( 0 で終了 ) : 25
素数生成中 ...
p = 32133097
べき指数を指定してください ( 0 で終了 ) : 1
e = 1

n = p^e = 32133097 ( 25 bits )

11946923 の mod n での平方根を求める

##### 反復 2 乗法を応用した方法 #####

平方根のリスト :
[9966497L, 22166600L]

検算 :
9966497 ^2
= 11946923

22166600 ^2
= 11946923

計算時間 = 0.0490000247955

##### 単純検索 #####

平方根 :
9966497
計算時間 = 423.071000099
```

奇素数べき p^e を法とする平方根の計算例

```
p = 11
e = 50
n = p^e = 1173908528796953165066664959903583199398982138987230 01  ( 173 bits )

3585231704213080237696663836338533924587303142606568 の mod n での平方根を求める

5 mod p
5 mod p^2
1215 mod p^3
13194 mod p^4
130322 mod p^5
774526 mod p^6
14947014 mod p^7
209818724 mod p^8
1281613129 mod p^9
8355456202 mod p^10
34292880803 mod p^11
1746162904469 mod p^12
17438304788074 mod p^13
328142714083453 mod p^14
4125641049915863 mod p^15
41720874574656722 mod p^16
363368983619661849 mod p^17
3901498183114718246 mod p^18
3901498183114718246 mod p^19
370856040873601995992 mod p^20
4407356010468962051198 mod p^21
4407356010468962051198 mod p^22
737032100492026812071087 mod p^23
879590428978916316228986 6 mod p^24
678943003446348297305609 12 mod p^25
( 中略 )
521210157595215759274924414691489562992480335175 51 mod p^48
246155483329265900731899872858170972727317684736113 mod p^49
344772419823004726000462249077703424379046692984238 6 mod p^50

平方根のリスト :
[344772419823004726000462249077703424379046692984238 6L,
829136108973948439066202710825879775010774696888061 5L]

検算 :
344772419823004726000462249077703424379046692984238 6 ^2
= 3585231704213080237696663836338533924587303142606568

829136108973948439066202710825879775010774696888061 5 ^2
= 3585231704213080237696663836338533924587303142606568

計算時間 = 0.201999902725
```

mod m での $X^2 \equiv a$ の解

$m \diagdown a$	0	1	2	3	4	5	6	7	8	9	10	11	12
2	0	1											
3	0	± 1	$-$										
4	$0, 2$	± 1	$-$	$-$									
5	0	± 1	$-$	$-$	± 2								
6	0	± 1	$-$	3	± 2	$-$							
7	0	± 1	± 3	$-$	± 2	$-$	$-$						
8	$0, 4$	$\pm 1, \pm 3$	$-$	$-$	± 2	$-$	$-$	$-$					
9	$0, \pm 3$	± 1	$-$	$-$	± 2	$-$	$-$	± 4	$-$				
10	0	± 1	$-$	$-$	± 2	5	± 4	$-$	$-$	± 3			
11	0	± 1	$-$	± 5	± 2	± 4	$-$	$-$	$-$	± 3	$-$		
12	$0, 6$	$\pm 1, \pm 5$	$-$	$-$	$\pm 2, \pm 4$	$-$	$-$	$-$	$-$	± 3	$-$	$-$	
13	0	± 1	$-$	± 4	± 2	$-$	$-$	$-$	$-$	± 3	± 6	$-$	± 5

法を奇素数 p に限ると

$p \diagdown a$	0	1	2	3	4	5	6	7	8	9	10	11	12
3	0	± 1	$-$										
5	0	± 1	$-$	$-$	± 2								
7	0	± 1	± 3	$-$	± 2	$-$	$-$						
11	0	± 1	$-$	± 5	± 2	± 4	$-$	$-$	$-$	± 3	$-$		
13	0	± 1	$-$	± 4	± 2	$-$	$-$	$-$	$-$	± 3	± 6	$-$	± 5

平方剰余記号

- 奇素数 p と整数 a を変数とする関数 $\left(\frac{a}{p}\right)$ を

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & a \text{ が } p \text{ で割り切れるとき} \\ 1 & a \text{ が } \texttt{mod } p \text{ の平方剰余であるとき} \\ -1 & a \text{ が } \texttt{mod } p \text{ の平方非剰余であるとき} \end{cases}$$

  で定める。(分数の記号と混同してはいけない !!)
- 平方剰余記号については「オイラーの規準」

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \quad \texttt{mod } p$$

  が成り立つので、反復 2 乗法を用いて高速に計算することができる。(ヤコビ記号を用いて更に高速に計算する方法もある。)
- 平方剰余記号は、
  - 素数判定
  - Rabin 暗号
  - コイントス・プロトコル
  - ゼロ知識証明
  - 楕円曲線への情報の埋め込み

  等、暗号理論に広範な応用を持つ。