暗号の数理要綱 #7

3.3 既約剰余類

合同類の積のところで述べたように，$\mathbb{Z}_m - \{0\}$ は，m が素数でなければ，積に関して群にならない。なぜなら，$m = pq$ となっているなら，$pq \equiv 0 \mod m$ であるから，$p, q \in \mathbb{Z}_m$ とみなした時に $pq = 0 \in \mathbb{Z}_m$ となってしまい，p, q は $\mathbb{Z}_m - \{0\}$ の中では積に関する逆元をもたないからである。(p^{-1} があれば $0 = p^{-1}pq = q$ となって $q \in \mathbb{Z}_m - \{0\}$ に矛盾する。)

しかし，$\{1, 2, \cdots, m-1\} = \mathbb{Z}_m - \{0\}$ の中には m と互いに素な数も存在する。今，$p \in \{1, 2, \cdots, m-1\}$ を $GCD(p, m) = 1$ となる数とすると，定理 3.2 より，ある整数 k_1, k_2 があって $k_1 p + k_2 m = 1$ となる。$k_1 p = -k_2 m + 1$ なので，

$$k_1 p \equiv 1 \mod m$$

となる。これは，k_1 を含む $\mathbb{Z}_m$ の剰余類 $[k_1]$ が $\mathbb{Z}_m - \{0\}$ において，積に関する p の逆元であることを示している。

定義 3.11 $m \geqq 2$ を自然数とする。

(1) $p \in \{1, \cdots, m-1\}$ が m と互いに素であるなら，p を含む $\mathbb{Z}_m$ の剰余類を **既約剰余類** という。

(2) $\mathbb{Z}_m$ における既約剰余類全体の集合を $\mathbb{Z}_m^*$ で表す。m が素数なら $\mathbb{Z}_m^* = \mathbb{Z}_m - \{0\}$ であるが，一般にはそうなるとは限らない。

(3) $\mathbb{Z}_m^*$ の元は積に関して逆元を持っているので，この集合は積に関して群をなす。これを **既約剰余類群** という。

(4) $\varphi(m) = |\mathbb{Z}_m^*|$ とおき，これを **オイラー関数** という。$\varphi(1) = 1$ と決める。

注意 1。 オイラー関数 $\varphi(m)$ は，$\{1, \cdots, m-1\}$ の中で m と互いに素であるものの個数を表す。例えば，p が素数ならば $\varphi(p) = p - 1$ である。また，

$$\varphi(4) = 2, \quad \varphi(6) = 2, \quad \varphi(8) = 4, \quad \varphi(9) = 6, \quad \varphi(10) = 4$$

など。

注意 2。 一般に，既約剰余類群は巡回群とは限らない。例えば，$\mathbb{Z}_8^* = \{[1], [3], [5], [7]\}$ であるが，$\mathbb{Z}_8^*$ において，

$$[3]^2 = 1 \qquad [5]^2 = 1 \qquad [7]^2 = 1$$

となって，単位元以外の全ての元の位数が 2 であり，どの元も $\mathbb{Z}_8^*$ 全体を生成しない。

定理 1.36 より次の定理が成り立つ。これは，フェルマーの小定理の一般形であり，オイラーによって証明された。

定理 3.12 (一般化されたフェルマーの小定理)　$m \geqq 2$ を自然数とする。

任意の $g \in \mathbb{Z}_m^*$ に対して

$$g^{\varphi(m)} = 1$$

が成り立つ。言い換えると，$a \in \mathbb{Z}$ が m と互いに素ならば,

$$a^{\varphi(m)} \equiv 1 \mod m$$

である。

特別なタイプの既約剰余類群が我々の考察対象である RSA 暗号をつくる群である。p, q を 2 つの素数とする。$m = pq$ としたとき既約剰余類群 $\mathbb{Z}_m^*$ が RSA 暗号の元になる群である。実際の RSA では p, q を十分大きく選ぶ必要があるのだが，ここでは p, q の小さい「**おもちゃの RSA**」を考える。Abel が おもちゃの RSA で暗号化したいと考えたとする。素数として，$p = 3, q = 5$ を選ぶ。$m = pq = 15$ なので

$$\mathbb{Z}_{15}^* = \{\, [1], [2], [4], [7], [8], [11], [13], [14] \,\}$$

である。暗号化の鍵である E と復号化の鍵である D を決める必要がある。暗号化の鍵としては $\varphi(15) = 8$ なので 8 と互いに素な数を選ぶ。今 $E = 3$ とする。D はある整数 k に対し

$$ED + \varphi(15)k = 1$$

が成立するような数を選ぶ。E から D を求めるアルゴリズムに関しては後で学ぶが，今は数が小さいので目の子でも求まる。例えば

$$3 \cdot 3 + 8 \cdot (-1) = 1$$

となるので $D = 3$ とする。平文の集合 $\mathcal{T}$ と $\mathbb{Z}_{15}^*$ の一対一対応が与えられているとする。とはいってもいまの場合文の種類は 8 通りしか考えられないが…。Abel はこの群 $\mathbb{Z}_{15}^*$ と暗号化鍵 3 を公開し，秘密鍵 3 は秘密裏に保管しておく。

Briskorn が Abel にある平文 T を送りたいとき，対応する $\mathbb{Z}_{15}^*$ の元が [2] だったとする。

$$[2]^E = [2]^3 = [8]$$

なので [8]，ないしは代表元の 8 を送る。

受け取った Abel は復号化鍵 3 を用いて

$$[8]^D = [8]^3 = ([8] \cdot [8])\,[8] = [64] \cdot [8] = [4] \cdot [8] = [32] = [2]$$

と計算してもとの [2] に対応する平文を得ることができる。

今 [2] に対しては $\left([2]^E\right)^D = [2]$ が成立したが，すべての元 g に対し $\left(g^E\right)^D = g$ が成立することが分かる。

演習問題 3.3　$\mathbb{Z}_{15}^*$ の任意の元 g に対し

$$\left(g^E\right)^D = g$$

が成立することを示せ。

この例では小さすぎてすべて計算することは手でも可能である。p, q を大きくしたとき，コンピュータを用いても解読されない安全性を持つかが問題になる。その議論のために次の節では整数に関するアルゴリズムを調べる。

3.4 ユークリッド互除法

2 つの自然数の最大公約数を計算するための効率的な方法として，**ユークリッド互除法** というものがある。これはその名のとおり，2300 年前に書かれた「ユークリッド原論」にすでに書かれてある。そのプロセスは以下のとおりである。

(1) まず，$n_0 > n_1$ を 2 つの自然数とする。$n_0 = n_1$ ならば最大公約数は $n_0 = n_1$ 自身なので考える必要はない。n_0 を n_1 で割った商を q_1 とし，余りを n_2 とすると，$n_0 = q_1\, n_1 + n_2$ であり，$0 \leqq n_2 \leqq n_1 - 1$ である。

(2) $n_2 = 0$ なら $n_0 = q_1\, n_1$ なので，$n_1 = GCD(n_0, n_1)$ である。

(3) $n_2 > 0$ なら，$GCD(n_0, n_1) = GCD(n_1, n_2)$ である。なぜならこの時，$n_0 = q_1\, n_1 + n_2$ なので n_0 は $GCD(n_1, n_2)$ の倍数であり，このことから $GCD(n_0, n_1) \geqq GCD(n_1, n_2)$ がわかる。また，$n_0 - q_1\, n_1 = n_2$ であるから，n_2 は $GCD(n_0, n_1)$ の倍数なので，$GCD(n_0, n_1) \leqq GCD(n_1, n_2)$ でなければならない。従って $GCD(n_0, n_1) = GCD(n_1, n_2)$ でなければならない。

(4) これにより，$GCD(n_0, n_1)$ を求める問題は，より小さな数のペアに対して $GCD(n_1, n_2)$ を求める問題へと置き換えられたことになる。あとは，このプロセスを繰り返せば良い。

(5) $GCD(n_0, n_1) = GCD(n_1, n_2)$ であるから，プロセスを繰り返しても，ペアの最大公約数は変わらない。

$n_i = q_{i+1}\, n_{i+1} + n_{i+2}$ となった時,

$n_{i+2} \neq 0$ **ならば：** $GCD(n_i, n_{i+1}) = GCD(n_{i+1}, n_{i+2})$ となり，その時はプロセスを続けることになる。

$n_{i+2} = 0$ **ならば：** $n_i = q_{i+1}\, n_{i+1}$ なので，$GCD(n_i, n_{i+1}) = n_{i+1}$ となり，n_{i+1} が最大公約数となる。

(6) $n_i > n_{i+1} > n_{i+2}$ なので，プロセスが進めば，数は必ず減少して行くので，このプロセスは有限回で終了する。

最後まで余りが 0 にならないなら，最後は余りが 1，すなわち $n_i = q_{i+1}\, n_{i+1} + 1$ となるが，これは，$GCD(n_i, n_{i+1}) = GCD(n_{i+1}, 1) = 1$ となり，最初の n_0 と n_1 が互いに素であることを示している。

例: $GCD(2397, 64515)$ を求めてみよう。

- $64515 = 26 \cdot 2397 + 2193$ であるから，
  $GCD(2397, 64515) = GCD(2193, 2397)$ である。
- $2397 = 1 \cdot 2193 + 204$ であるから，
  $GCD(2193, 2397) = GCD(204, 2193)$ である。
- $2193 = 10 \cdot 204 + 153$ であるから，
  $GCD(204, 2193) = GCD(153, 204)$ である。
- $204 = 1 \cdot 153 + 51$ であるから，
  $GCD(153, 204) = GCD(51, 153)$ である。

- $153 = 3 \cdot 51$ であるから，$GCD(51, 153) = 51$ である。
- 以上から，$GCD(2397, 64515) = 51$ がわかった。

このアルゴリズムの計算量を考えてみよう。

Case 1: $n_1 > \dfrac{n_0}{2}$ の場合。

この時は明らかに $q_1 = 1$ であり，$n_0 = n_1 + n_2$ となるので，

$$n_2 = n_0 - n_1 < n_0 - \frac{n_0}{2} = \frac{n_0}{2}$$

となる。

Case 2: $n_1 \leqq \dfrac{n_0}{2}$ の場合。

この時も，$n_2 < n_1 \leqq \dfrac{n_0}{2}$ となる。

以上から，どちらの場合であっても，$n_2 < \dfrac{n_0}{2}$ となる。すなわち，このプロセスを 2 回行うことにより，n_{i+2} は n_i の 1/2 倍以下となる。

n_0 が大体 $n_0 = 2^\alpha$ 位の数であるなら，n_2 は $2^{\alpha-1}$ 以下の数であり，n_4 は $2^{\alpha-2}$ 以下の数であり，これが続いて行く。従って，この操作を 2α-回 行えば，$n_{2\alpha}$ は 1 以下となり，どんな場合でも，操作は終了する。

$\alpha = \log_2 n_0$ であるから，多くとも $2\log_2 n_0$ 回の操作で，最初の数の最大公約数に到達する。

例えば，RSA 暗号などで使われる 200 桁くらいの数を考えてみよう。まず，200 桁程度の割り算はほとんど計算時間はかからない。また，10 進数で 200 桁くらいの数は 2 進数で見ると 660 桁くらいなので，これを大きな方とする 2 つの数についてユークリッド互除法を適用すると，多くても 1300 回程度のプロセスで最大公約数が見つかることになる。これは現在の計算機にとっては，極めて短時間で計算できる程度のものである。このアルゴリズムは，以下の既約剰余類の逆元の計算に使われるなど，RSA 暗号を実現する上で最も重要なものである。

演習問題 3.4　2 つの自然数 m, n を入力すると，その最大公約数 $GCD(m, n)$ を出力するプログラムを作れ。可能なものは BigInteger クラスを用いていくらでも大きい自然数に対応可能なものを作れ。

3.5　拡張ユークリッド・アルゴリズム

前前セクションで見たように，$m \geqq 2$ を自然数とすると，$\mathbb{Z}_m^*$ の任意の元 e は積に関する逆元 e^{-1} を持つ。(なお，ここで e は積に関する単位元ではなく，単なる $\mathbb{Z}_m^*$ の元の意味で使っているので注意。)

それでは具体的に，m と互いに素な $e \in \{1, 2, \cdots, m-1\}$ が与えられた時，それの $\mathbb{Z}_m^*$ における逆元 $e^{-1} \in \{1, 2, \cdots, m-1\}$ はどのように計算したら良いのだろうか？ 実はこの計算は，RSA 暗号において，その根幹をなす重要な部分となる。

m と e は互いに素なので，定理 3.2 より，ある整数 k_1, k_2 があって，$k_1e + k_2m = 1$ となる。$k_1e = -k_2m + 1$ なので，$k_1e \equiv 1 \mod m$ であり，これは，k_1 を含む剰余類が e の逆元であることを示している。

従って，$e \in \mathbb{Z}_m^*$ の逆元を求める問題は，

- $k_1e + k_2m = 1$ を満たす k_1 を求める問題である。

と言える。k_1 が求まれば自動的に k_2 も決まってしまうので，これは，

- $k_1e + k_2m = 1$ を満たす (k_1, k_2) を求める問題。

と言うことができる。

この (k_1, k_2) は，実はユークリッドの互除法を利用することにより求めることができる。この方法を **拡張ユークリッド・アルゴリズム** という。それは，以下のようにして行う。

$e \in \{1, 2, \cdots, m-1\}$ であるので $e < m$ である。$GCD(m, e)$ を求めるユークリッドの互除法を行ってみる。$GCD(m, e) = 1$ なので，これは最後の余りが 1 になった時点で終了する。

$$\begin{aligned}
m &= q_1e + r_1 \\
e &= q_2r_1 + r_2 \\
r_1 &= q_3r_2 + r_3 \\
r_2 &= q_4r_3 + r_4 \\
\cdots &\quad \cdots \\
r_{s-2} &= q_sr_{s-1} + r_s \\
r_{s-1} &= q_{s+1}r_s + r_{s+1} \\
r_s &= q_{s+2}r_{s+1} + 1
\end{aligned}$$

となったとする。

- $m = q_1e + r_1$ より $r_1 = m - q_1e$ である。すなわち，最初の余り r_1 は m と e の 1 次結合で表されている。
- $e = q_2r_1 + r_2$ より $r_2 = e - q_2r_1$ である。すなわち，r_2 は e と r_1 の 1 次結合で表されている。しかし，最初の式から，r_1 は m と e の 1 次結合で表されているので，結局，r_2 は m と e の 1 次結合で表される。
- $r_1 = q_3r_2 + r_3$ より $r_3 = r_1 - q_3r_2$ である。すなわち，r_3 は r_1 と r_2 の 1 次結合で表されている。しかし，r_1 と r_2 は共に m と e の 1 次結合で表されているので，結局，r_3 は m と e の 1 次結合で表される。
- このプロセスを繰り返すことにより，全ての余り r_i は，m と e の 1 次結合で表されることになる。
- 最後の余りは 1 なので，最終的に，1 が m と e の 1 次結合で表されることになる。これにより，$k_1e + k_2m = 1$ という式に到達する。

この方法を見るとわかるように，基本的にこれはユークリッド互除法であり，アルゴリズムの計算量はユークリッド互除法とほとんど同じである。すなわち，200 桁程度の m や e に対しては，現在の計算機の能力からすれば，極めて短時間で計算可能である。

例: 具体例で見てみよう。

$$36\,a + 47\,b = 1$$

を満たす整数 a, b を求めることを考える。

47 は素数なので，当然 $GCD(36, 47) = 1$ であり，定理 3.3 より，求める整数解 a, b は存在する。

ユークリッド互除法により，36, 47 の最大公約数 1 を求めるプロセスを実行してみると，

$$\begin{aligned} 47 &= 36 + 11 \\ 36 &= 3 \cdot 11 + 3 \\ 11 &= 3 \cdot 3 + 2 \\ 3 &= 2 + 1 \end{aligned}$$

となる。

拡張ユークリッド・アルゴリズムは，ユークリッド互除法を実行しながら，この余りの数 11, 3, 2, 1 を 36 と 47 の 1 次結合で表して行く，というプロセスである。

具体的には，

$$\begin{aligned} 11 &= -36 + 47 \\ 3 &= 36 - 3 \cdot 11 = 36 - 3 \cdot (-36 + 47) = 4 \cdot 36 - 3 \cdot 47 \\ 2 &= 11 - 3 \cdot 3 = (-36 + 47) - 3 \cdot (4 \cdot 36 - 3 \cdot 47) = -13 \cdot 36 + 10 \cdot 47 \\ 1 &= 3 - 2 = (4 \cdot 36 - 3 \cdot 47) - (-13 \cdot 36 + 10 \cdot 47) = 17 \cdot 36 - 13 \cdot 47 \end{aligned}$$

となり，$a = 17, b = -13$ が得られた。

ちなみに，この $17 \cdot 36 - 13 \cdot 47 = 1$ より，

$$36 \cdot 17 \equiv 1 \mod 47$$

がわかる。すなわち，$\mathbb{Z}_{47}^*$ において，36 の積に関する逆元は 17 である。

演習問題 3.5 互いに素である 2 つの自然数 m, n を入力すると，$\mathbb{Z}_m^*$ における n の逆元 n^{-1} を出力するプログラムを作れ。可能なものは BigInteger クラスを用いていくらでも大きい自然数に対応可能なものを作れ。