Tonelli–Shanks Algorithm

ナンブ キトラ

2021 年 1 月 28 日

この文章では，有限素体で平方剰余になっている元の平方根を求めるアルゴリズムである Tonelli–Shanks のアルゴリズムと，このアルゴリズムの n 乗根への一般化を解説する．

1　オイラーの規準

p を素数とし，$r \in \mathbb{Z}_{\geq 0}$ とする．このとき $a \in \mathbb{Z}$ が**法 p における r 乗剰余**であるとはある $b \in \mathbb{Z}$ が存在して

$$b^r \equiv a \mod p$$

となるときにいう．$r = 2, 3$ のときはそれぞれ平方剰余と立方剰余と呼ばれる．

このセクションでは法 p において $a \in \mathbb{Z}$ が r 乗剰余になるための必要十分条件を与えるオイラーの規準を紹介する．

まず重要なフェルマーの小定理を紹介する．証明は省略する．

補題 1. p を素数とし，$a \in \mathbb{Z}$ とする．このとき

$$a^{p-1} \equiv 1 \mod p$$

が成り立つ．

以下の命題がオイラーの規準である．通常は $r = 2$ の場合の命題をそのように呼ぶ．

命題 2. p を素数とし，$r \in \mathbb{Z}_{\geq 0}$ とする．このとき $a \in \mathbb{Z}$ が法 p における r 乗剰余であることと，

$$a^{\frac{p-1}{\gcd(p-1,r)}} \equiv 1 \mod p$$

が成り立つことは同値である．

証明. まず最初に a が法 p における r 乗剰余であると仮定する．このとき $b^r \equiv a \mod p$ となる $b \in \mathbb{Z}$ が存在する．するとフェルマーの小定理より

$$a^{\frac{p-1}{\gcd(p-1,r)}} \equiv b^{\frac{r}{\gcd(p-1,r)}(p-1)} \equiv 1 \mod p$$

が成り立ち，定理の前半が証明された．後半を示そう．

まず最初に $\gcd(p-1, r) = 1$ の場合に証明をする．このとき $(p-1)/\gcd(p-1, r) = p-1$ なのでフェルマーの小定理から，任意の $a \in \mathbb{Z}$ が法 p における r 乗剰余であることを示せば良い．さて今の状況で r と $p-1$ が互いに素なので，$h \in \mathbb{Z}_{\geq 0}$ が存在して

$$hr \equiv 1 \mod p-1$$

となる．このときフェルマーの小定理より a^h が法 p における a の r 乗根になる．

次に $\gcd(p-1,r) \neq 1$ の場合を証明する．g を法 p における原始根とし，$k \in \{0,1,\ldots,p-1\}$ を用いて $g^k \equiv a \mod p$ となっているとする．このとき

$$a^{\frac{p-1}{\gcd(p-1,r)}} \equiv g^{k\frac{p-1}{\gcd(p-1,r)}} \equiv 1 \mod p$$

である．g の法 p における位数は $p-1$ なので，$k\frac{p-1}{\gcd(p-1,r)}$ は $p-1$ の倍数である．よって $m = \frac{k}{\gcd(p-1,r)}$ は整数であり，特に $k = m\gcd(p-1,r)$ とかける．さて，今 $r/(\gcd(p-1,r))$ は $p-1$ と互いに素なので，ある $h \in \mathbb{Z}_{\geq 0}$ が存在して $hr/(\gcd(p-1,r)) \equiv 1 \mod p-1$ となる．上で得られた m と h を使って，$b = g^{mh}$ と定義する．するとフェルマーの小定理から

$$b^r \equiv g^{mhr} \equiv g^{m\gcd(p-1,r)\cdot h\frac{r}{\gcd(p-1,r)}} \equiv g^{k\cdot h\frac{r}{\gcd(p-1,r)}} \equiv a^{h\frac{r}{\gcd(p-1,r)}} \equiv a \mod p$$

となるので，b が求めている a の法 p における r 乗根である． □

注意. 命題 *2* の証明の後半は a の r 乗根を求める手順になっている．しかし原始根を使っているので p が大きいときには *(*証明には使えるが*)* とても計算には使えない手順になっている．法 p における冪乗根の計算では如何に原始根を用いずに計算するかが重要になっていくる．

2 Tonelli–Shanks のアルゴリズム

このセクションでは有限素体における平方根を求めるアルゴリズムの一つである Tonelli–Shanks のアルゴリズムの理論的背景と，アルゴリズムを紹介する．

以下の補題が Tonelli-Shanks のアルゴリズムの理論的支柱である．

補題 3. p を奇素数とし，$s \in \mathbb{N}$ と奇数 d は $p-1 = 2^s d$ を満たすとする．z を法 p における平方非剰余とする．$1 \leq i < s$ として，$t \in \mathbb{Z}$ は法 p における 1 の 2^{s-i} 乗根であり，なおかつ法 p における 1 の 2^{s-i-1} 乗根ではないとする．このとき $b = z^{2^{i-1}d}$ とすると tb^2 は 1 の 2^{s-i-1} 乗根である．

証明. t は法 p における 1 の 2^{s-i} 乗根であるということは，$t^{2^{s-i-1}}$ は法 p における 1 の平方根である．仮定からこの値は法 p で -1 に合同である．今，オイラーの規準から，

$$z^{\frac{p-1}{2}} \equiv z^{2^{s-1}d} \equiv -1 \mod p$$

であることに注意すると，

$$(tb^2)^{2^{s-i-1}} \equiv t^{2^{s-i-1}} z^{2^{s-1}d} \equiv (-1)(-1) \equiv 1 \mod p$$

なので，tb^2 は法 p で 1 の 2^{s-i-1} 乗根である． □

この補題 3 を用いて法 p における平方剰余の平方根を求めるアルゴリズムが Tonelli–Shanks のアルゴリズムである．

考察 4. p を素数とし，a を法 p における平方剰余とする．このとき a の法 p における平方根を求めたい．

最初に $p-1 = 2^s d$ となるような s と奇数 d を求める．

次に平方非剰余 z を適当に取ってくる．1 から $p-1$ までの間にはこのような数は $(p-1)/2$ 個あるので，結構な確率で取ってこれる．

次に $R = a^{\frac{1+d}{2}}$ とおく．d は奇数なので $1 + d/2$ はちゃんと整数になっている．すると

$$R^2 = a^{1+d} = a \cdot a^d \mod p$$

となる．$t = a^d$ とおく．もしも $t \equiv 1 \mod p$ ならば R が a の法 p における平方根になっている．しかし一般にはこういう風にはなっていない．一般の場合も R の値を修正して，$t \equiv 1$ となる理想的なケースを目指すことを考える．まず，a は平方剰余なので，オイラーの規準から

$$t^{2^{s-1}} \equiv a^{\frac{p-1}{2}} \equiv 1 \mod p$$

なので，t は法 p における 1 の 2^{s-1} 乗根になっている．t が法 p における 1 の 2^{s-s} 乗根，すなわち 1 乗根ならば，t は 1 に合同なので，R が a の法 p における平方根である．t がそうではないとすると，次のような i を見つけることができる: $1 \leq i < s$ であり，t は法 p における 1 の 2^{s-i} 乗根であるが，2^{s-i-1} 乗根ではない．よって，補題 *3* を適応することによって，$b = z^{2^{i-1}d}$ とすると，tb^2 は法 p における 1 の 2^{s-i-1} 乗根である．よって，$R' = Rb$ と値を修正すると，

$$(R')^2 \equiv a \cdot tb^2 \mod p$$

となる．$t' = tb^2$ とすると，t' は法 p における 1 の 2^{s-i-1} 乗根になっている．

以上の操作は t が法 p における 1 の 2^{s-s} 乗根，すなわち，$t \equiv 1 \mod p$ となるまで続けることができ，そのときの R が a の法 p における平方根である．

以上の操作を疑似コードにして表したのが次である．

Algorithm 1 Solving $x^2 \equiv a \mod p$

Require: a:integer and p:prime

```
if p = 2 then
  return a
end if
Take s ≥ 1 and an odd number d with p − 1 = 2^s d
Take z ∈ ℤ s.t. z^{(p−1)/2} ≢ 1 mod
M ← s
R ← a^{(1+d)/2}
t ← a^d
c ← z^d
while M ≠ 0 do
  Take the least i ∈ {1, ..., s − 1} such that t^{2^i} ≡ 1 mod p
  b ← c^{2^{M−i−1}}
  M ← i
  c ← b^2
  R ← Rb
  t ← tb^2
end while
return R
```

考察では $t^{2^{s-i}} \equiv 1$ となる最大の i を取ってきたが，$t^{2^i} \equiv 1$ となる最小の i を見つける方が計算量が少なくて済むのでそのように修正している．このコードは wikipedia の Tonelli–Shanks のページを大いに参考にした．

以上で説明したアルゴリズムはある種伝統的なアルゴリズムであったが，次の補題を利用しても平方根を求めることができる．

補題 5. p を奇素数とし，$s \in \mathbb{Z}_{\geq 2}$ と奇数 d は $p-1 = 2^s d$ を満たすとする．z を法 p における平方非剰余とする．そして a を法 p における平方剰余とする．このとき，$m \in \mathbb{Z}$ が存在して $a^d z^{2md} \equiv 1 \mod p$ となる．この m を用いると，$a^{\frac{1+d}{2}} z^{md}$ は法 p における a の平方根の一つである．

証明. 補題の後半は自乗すれば直ちにわかる．前半を示す．もし $s = 1$ ならば，オイラーの規準より $m = 0$ とすれば条件を満たすので，以下では $s \geq 2$ とする．$\{0,1\}$ に値をとる点列 $\{c_i\}_{i=0}^{s-2}$ を任意の $n \in \{0, 1, \ldots, s-2\}$ について

$$a^{2^{s-n-2}d} \cdot z^{(2^{s-n-1}c_0 + 2^{s-n}c_1 + \cdots + 2^{s-2}c_{n-1} + 2^{s-1}c_n)d} \equiv 1 \mod p$$

を満たすように帰納的に以下のように定める．まず，c_0 を定義する．オイラーの規準から $a^{2^{s-1}d} \equiv 1 \mod p$ である．よって $a^{2^{s-2}d}$ は 1 か -1 に合同である．$a^{2^{s-2}d}$ が -1 に合同ならば $c_0 = 1$，そうでない時は $c_0 = 0$ とする．

一般に n に対して n 以下の自然数に対して c_i が定義されたとしよう．このとき，帰納法の仮定から，

$$a^{2^{s-n-1}d} \cdot z^{(2^{s-n}c_0+2^{s-n+1}c_1+\cdots+2^{s-1}c_{n-1})d} \equiv 1 \mod p$$

である．よって，その平方根である $a^{2^{s-n-2}d} z^{(2^{s-n-1}c_0+2^{s-n}c_1+\cdots+2^{s-2}c_{n-1})d}$ は 1 か -1 に合同である．この値がが -1 に合同ならば $c_n = 1$, そうでないときは $c_n = 0$ と定義する．

以上によって $\{c_i\}_{i=1}^{s-2}$ が得られたが，帰納法の仮定と，c_{s-2} の定義より，

$$a^d z^{(2c_0+2^2c_1+\cdots+2^{s-1}c_{s-2})d} \equiv 1 \mod 1$$

このとき，$m = c_0 + 2c_1 + \cdots + 2^{s-2}c_{s-2}$ とすると補題の条件を満たす． □

この補題を使ったアルゴリズムの疑似コードを以下に書く．このコードは [2] を大いに参考にした．このアルゴリズムはプログラミングの初心者でも書きやすいという利点がある．

Algorithm 2 Solving $x^2 \equiv a \mod p$

Require: a:integer and p:prime
  if $p = 2$ **then**
    return a
  end if
  if $a^{\frac{p-1}{2}} \not\equiv 1 \mod p$ **then**
    return Error
  end if
  Take $z \in \{1, \ldots, p-1\}$ s.t. $z^{\frac{p-1}{2}} \not\equiv 1 \mod p$
  $apow \leftarrow p - 1$
  $zpow \leftarrow 0$
  while $apow$ is even **do**
    $apow \leftarrow apow/2$
    $zpow \leftarrow zpow/2$
    if $a^{apow} z^{zpow} \not\equiv 1 \mod p$ **then**
      $zpow \leftarrow zpow + (p-1)/2$
    end if
  end while
  $apow \leftarrow (apow + 1)/2$
  $zpow \leftarrow zpow/2$
  return $a^{apow} z^{zpow}$

3 一般の冪根を求めるアルゴリズム (Adleman-Manders-Miler)

このセクションでは一般の $r \in \mathbb{Z}_{\geq 1}$ に対して $\mathbb{F}_p$ における r 乗根を求めることを考える．まずフェルマーの小定理から次の命題がわかる．オイラーの規準の証明の中でも用いていた手法である．証明は省略する．

命題 6. p を素数とし，$r \in \mathbb{Z}_{\geq 1}$ とする．そして $\gcd(p-1, r) = 1$ と仮定する．このとき法 $p-1$ にける r の逆数を h とすると，任意の $a \in \mathbb{Z}$ の法 p における r 乗根は a^h である．

よって以下の考察では，r は $p-1$ の素因数と仮定する命題が多くなる．

補題 7. p を素数とし，r を $p-1$ の素因数であるとする．$s \in \mathbb{N}$ と r で割り切れない数 d は $p-1 = r^s d$ を満たすとする．z を法 p における r 乗非剰余とする．$1 \leq i < s$ として，$t \in \mathbb{Z}$ は法 p における 1 の r^{s-i} 乗根であり，なおかつ法 p における 1 の r^{s-i-1} 乗根ではないとする．$b = z^{r^{i-1}d}$ とする．このときある $m \in \{0, 1, \ldots, p-1\}$ が存在して tb^{mr} が 1 の r^{s-i-1} 乗根となる．

証明. t が法 p における 1 の r^{s-i} 乗根であるということは，$t^{r^{s-i-1}}$ は法 p における 1 の r 乗根である．今オイラーの規準から，$\gamma = z^{\frac{r^{s-1}}{d}}$ は 1 とは異なる 1 の r 乗根であり，r は素数なので，$\{\gamma^m\}_{m=0}^{r-1}$ は 1 の r 乗根を全て含む．よって，ある m が存在して

$$t^{r^{s-i-1}} \cdot \gamma^m = 1$$

を満たす．よって

$$(tb^{mr})^{r^{s-i-1}} \equiv t^{r^{s-i-1}} z^{r^{s-1}md} \equiv t^{r^{s-i-1}} \cdot \gamma^m \equiv 1 \mod p$$

なので，tb^{mr} は法 p で 1 の r^{s-i-1} 乗根である． □

考察 8. p を素数とし，r を自然数とする．a を法 p における r 乗剰余とする．このとき a の法 p における r 乗根を求めたい．

まず最初に，$r/\gcd(p-1, r)$ の法 $p-1$ における逆数を h として a^h とするとこれは a の $r/\gcd(p-1, r)$ 乗根になっているので，a^h の $\gcd(p-1, r)$ 乗根を求めれば a の r 乗根になる．よって以下では r を $p-1$ の約数とする．

次に r を素因数分解し，$r = q_1 q_2 \ldots q_l$ とする．ここで，q_i たちには重複があっても良い．このとき a の q_1 乗根を求めそれを $f_1(a)$ とする．そして $f_1(a)$ の q_2 乗根を $f_2(f_1)(a)$ とすると，a の r 乗根は $f_l \circ f_{l-1} \circ \ldots f_2 \circ f_1(a)$ である．よって a の r 乗根を求めるには r を素数としても良い．

次に $p-1 = r^s d$ となるような s と r で割り切れない d を求める．r は $p-1$ の素因数なので $s \geq 1$ に注意せよ．

次に r 乗非剰余 z を適当に取ってくる．1 から $p-1$ までの間にはこのような数は $(r-1)(p-1)/r$ 個あるので，結構な確率で取ってこれる．

さて，次に平方根のときの同様に $R = a^{\frac{1+d}{r}}$ とおきたいところだが，$d+1$ が r の倍数とは全く限らない．そこで平方根の場合の議論ではなく，$\gcd(r, p-1) = 1$ の場合の議論を参考にして R の値を定める．つまり今の場合には r の法 d における逆数を h として $R = a^h$ とする．

$$R^r = a^{rh} = a \cdot a^{rh-1} \mod p$$

となる．$t = a^{rh-1}$ とおく．もしも $t \equiv 1 \mod p$ なら R が a の法 p における r 乗根になっている．しかし一般にはこういう風にはなっていない．一般の場合も R の値を修正して，$t \equiv 1$ となる理想的なケースを目指すことを考える．

h の定義から $rh-1$ は d の倍数になっているので $r^{s-1}(rh-1)$ が $\frac{p-1}{r}$ の倍数になっていることに注意すると a は r 乗剰余なので，オイラーの規準から $t^{r^{s-1}} \equiv a^{r^{s-1}(rh-1)} \equiv 1 \mod p$ がわかる．よって t は法 p における 1 の r^{s-1} 乗根になっている．t が法 p における 1 の r^{s-s} 乗根，すなわち 1 乗根ならば t は 1 に合同

なので R が a の法 p における平方根である．t がそうではないとすると，次のような i を見つけることができる: $1 \leq i < s$ であり，t は法 p における 1 の r^{s-i} 乗根であるが，r^{s-i-1} 乗根ではない．よって，補題 7 にあるような $m \in \{0, 1, \ldots, p-1\}$ を見つけることができる．$b = z^{r^{i-1}d}$ とすると，tb^{mr} は法 p における 1 の r^{s-i-1} 乗根である．よって，$R' = Rb^m$ と値を修正すると，

$$(R')^r \equiv a \cdot tb^{mr} \mod p$$

となる．$t' = tb^{mr}$ とすると，t' は法 p における 1 の r^{s-i-1} 乗根になっている．

以上の操作は t が法 p における 1 の r^{s-s} 乗根，すなわち，$t \equiv 1 \mod p$ となるまで続けることができ，そのときの R が a の法 p における r 乗根である．

以上の操作を疑似コードにして表したのが次である．ここでは r は $p-1$ の素因数で a は r 乗剰余であることがわかっているとする．

Algorithm 3 Solving $x^r \equiv a \mod p$

Require: a:integer and p:prime

Take $s \geq 1$ and a number d s.t. $p - 1 = r^s d$ and $\gcd(d, q) = 1$
Take h s.t. $rh \equiv 1 \mod d$
Take $z \in \mathbb{Z}$ s.t. $z^{\frac{p-1}{r}} \not\equiv 1 \mod$
$M \leftarrow s$
$R \leftarrow a^h$
$t \leftarrow a^{rh-1}$
$c \leftarrow z^d$
while $M \neq 0$ **do**
  Take the least $i \in \{1, \ldots, s-1\}$ such that $t^{r^i} \equiv 1 \mod p$
  Take $m \in \{0, \ldots, p-1\}$ s.t. $t^{r^{i-1}} \left(z^{(p-1)/r}\right)^m \equiv 1 \mod p$
  $b \leftarrow c^{r^{M-i-1}}$
  $M \leftarrow i$
  $c \leftarrow b^r$
  $R \leftarrow Rb^m$
  $t \leftarrow tb^{mr}$
end while
return R

ここでもやはり，$t^{r^{s-i}} \equiv 1$ となる最大の i ではなく $t^{r^i} \equiv 1$ となる最小の i を見つけるように修正をしている．また，アルゴリズムの中の “Take $m \in \{0, \ldots, p-1\}$...” の部分は離散対数問題を解く必要があるので r が大きすぎると計算機でも時間がかかる計算になる．この一般の r 乗根を求めるアルゴリズムは Adleman-Manders-Miler のアルゴリズムとして知られているようである．

次の補題を利用しても r 乗根を求めることができる．この補題は [1] を大いに参考にした．

補題 9. p を素数とし，r を $p-1$ の素因数とする．$s \in \mathbb{N}$ と r で割り切れない数 d は $p - 1 = r^s d$ を満たすとする．z を法 p における r 乗非剰余とする．そして a を法 p における r 乗剰余とする．h を法 d における r の逆数とする．このとき，$m \in \mathbb{Z}$ が存在して $a^{rh-1} z^{rmd} \equiv 1 \mod p$ となる．この m を用いると，$a^h z^{md}$

は法 p における a の r 乗根の一つである．

証明. 補題の後半は r 乗すれば直ちにわかる．前半を示す．もし $s = 1$ ならば，オイラーの規準より $m = 0$ とすれば条件を満たすので，以下では $s \geq 2$ とする．仮定より z は r 乗非剰余なので $\gamma = z^{r^{s-1}d}$ とすると，γ は法 p における 1 の r 剰余でありなおかつ 1 とは異なることに注意せよ．また，r が素数なので $\{\gamma^k\}_{k=0}^{r-1}$ は法 p における r 乗根を全て含んでいることに注意しよう．$\{0, 1, \ldots, r-1\}$ に値をとる点列 $\{c_i\}_{i=0}^{s-2}$ を任意の $n \in \{0, 1, \ldots, s-2\}$ について

$$\left(a^{rh-1}\right)^{r^{s-n-2}} \cdot z^{(r^{s-n-1}c_0 + r^{s-n}c_1 + \cdots + r^{s-2}c_{n-1} + r^{s-1}c_n)d} \equiv 1 \mod p$$

を満たすように帰納的に以下のように定める．まず，c_0 を定義する．オイラーの規準から $\left(a^{rh-1}\right)^{r^{s-1}} \equiv 1 \mod p$ である．よって $\left(a^{rh-1}\right)^{r^{s-2}}$ は法 p における 1 の r 乗根である．ここで c_0 を $\left(a^{rh-1}\right)^{r^{s-2}} \cdot \gamma^{c_0} \equiv 1$ を満たすようにとる．一般に n に対して n 以下の自然数に対して c_i が定義されたとしよう．このとき，帰納法の仮定から，

$$\left(a^{rh-1}\right)^{r^{s-n-1}} \cdot z^{(r^{s-n}c_0 + r^{s-n+1}c_1 + \cdots + r^{s-1}c_{n-1})d} \equiv 1 \mod p$$

である．よって，その r 乗根である $\left(a^{rh-1}\right)^{r^{s-n-2}d} z^{(r^{s-n-1}c_0 + r^{s-n}c_1 + \cdots + r^{s-2}c_{n-1})d}$ 法 p における 1 の r 乗根である．c_n を $\left(a^{rh-1}\right)^{r^{s-n-2}d} z^{(r^{s-n-1}c_0 + r^{s-n}c_1 + \cdots + r^{s-2}c_{n-1})d} \cdot \gamma^{c_n} \equiv 1 \mod p$ となるように定義する．以上によって $\{c_i\}_{i=1}^{s-2}$ が得られたが，帰納法の仮定より $a^{rh-1} z^{(rc_0 + r^2c_1 + \cdots + r^{s-1}c_{s-2})d} \equiv 1 \mod p$ が成り立つ．このとき，$m = c_0 + rc_1 + \cdots + r^{s-2}c_{s-2}$ とすると補題の条件を満たす． □

この補題を用いたアルゴリズムの疑似コードを書くと以下のようになる．ここでは r は $p-1$ の素因数で a は r 乗剰余であることがわかっているとする．

Algorithm 4 Solving $x^r \equiv a \mod p$

Require: p:prime, r:prime s.t. $r|p-1$, and a:integer s.t. $a^{\frac{p-1}{r}} \equiv 1 \mod p$
  Take $z \in \{1, \ldots, p-1\}$ s.t. $z^{\frac{p-1}{r}} \not\equiv 1 \mod p$
  Take s and d s.t. $p - 1 = r^s d$ and $\gcd(r, d) = 1$
  Take h s.t. $rh \equiv 1 \mod d$
  $apow \leftarrow (rh-1)r^s$
  $zpow \leftarrow 0$
  while $apow \equiv 0 \mod r$ **do**
    $apow \leftarrow apow/r$
    $zpow \leftarrow zpow/r$
    if $a^{apow} z^{zpow} \not\equiv 1 \mod p$ **then**
      Take $m \in \{0, \ldots, p-1\}$ s.t. $(a^{apow} z^{zpow}) \cdot \left(z^{(p-1)/r}\right)^m \equiv 1 \mod p$
      $zpow \leftarrow zpow + m(p-1)/r$
    end if
  end while
  $apow \leftarrow (apow + 1)/r$
  $zpow \leftarrow zpow/r$
  return $a^{apow} z^{zpow}$

一般の $r \in \mathbb{Z}_{\geq 1}$ に対して r 乗根を求めるプログラムは以下のようになる．以下に現れる $Solve\_Root\_prime(x, r, p)$ は r が $p-1$ の素因数のときの x の法 p における r 乗根を表すとする．このような値を求めるプログラムは既に上で記述している．

Algorithm 5 Solving $x^r \equiv a \mod p$

Require: a:integer, r:positive integer, p:prime

if $a^{(p-1)/\gcd(p-1,r)} \not\equiv 1 \mod p$ **then**
  return 'a is not an r-th residue.'
end if
$x \leftarrow a$
if $\gcd(p-1, r) = 1$ **then**
  Take v s.t. $vr \equiv 1 \mod p-1$
  return x^v
else
  Take v s.t. $v \cdot \left(\frac{r}{\gcd(p-1,r)}\right) \equiv 1 \mod p-1$
  $x \leftarrow x^v$
end if
Take primes $q_1, \ldots, q_l$ s.t. $\gcd(p-1, r) = q_1 \cdot q_2 \cdots q_l$
for $i \in \{1, \ldots, l\}$ **do**
  $x \leftarrow Solve\_Root\_prime(x, q_i, p)$
end for
return x

参考文献

[1] Z. Cao, Q. Sha and X. Fan, *Adleman-Manders-Miler root extraction method revised*, in Information Security and Cryptology (Chuan-Kun Wu, Moti Yung and Dongdai Lin, eds.), 7th International Conference, Inscrypt 2011 Beijing, China, November/December 2011 Revised Selected Papers, (2012), pp77–85.

[2] R. Kumar, *A simple algorithm for finding square rot modulo p*, (2020), arXiv:2008.11814