

アルゴリズム論特論 (塩田)

ユークリッドのアルゴリズム

ユークリッドのアルゴリズム

入力：整数 a, b

出力： $\gcd(a, b)$

Step 1: 数列 $\{r_n\}$ を宣言。

Step 2: $r_0 \leftarrow a, r_1 \leftarrow b, n \leftarrow 0$.

Step 3: $r_{n+1} = 0$ ならば $\text{abs}(r_n)$ を出力。

Step 4: $r_{n+2} \leftarrow (r_n \% r_{n+1}), n \leftarrow n + 1$ として Step 3 へ。

拡張ユークリッドアルゴリズム

入力：整数 a, b

出力： $\gcd(a, b)$ と $\gcd(a, b) = ax + by$ を満たす整数 x, y

Step 1: 数列 $\{r_n\}, \{x_n\}, \{y_n\}$ を宣言。

Step 2: $(r_0, x_0, y_0) \leftarrow (a, 1, 0), (r_1, x_1, y_1) \leftarrow (b, 0, 1), n \leftarrow 0$.

Step 3: $r_{n+1} = 0$ ならば Step 5 へ。

Step 4: $q \leftarrow \lfloor r_n / r_{n+1} \rfloor,$
 $r_{n+2} \leftarrow r_n - q \times r_{n+1},$
 $x_{n+2} \leftarrow x_n - q \times x_{n+1},$
 $y_{n+2} \leftarrow y_n - q \times y_{n+1},$
 $n \leftarrow n + 1$ として Step 3 へ。

Step 5: $r_n < 0$ ならば $(r_n, x_n, y_n) \leftarrow (-r_n, -x_n, -y_n)$.

Step 6: (r_n, x_n, y_n) を出力。

アルゴリズムの根拠

- (1) $c = a \% b$ のとき $\gcd(a, b) = \gcd(b, c)$ が成立すること。
- (2) r_n は減少列ゆえ Step 3 の終了条件がかならず達成されること。
- (3) $\gcd(a, 0) = \text{abs}(a)$ であること。
- (4) 常に $r_n = a \times x_n + b \times y_n$ が成立すること。

実行例

```
a = 1234567890
b = 135792468
r[ 0] = 1234567890,  x[ 0] = 1,      y[ 0] = 0,
r[ 1] = 135792468,  x[ 1] = 0,      y[ 1] = 1,
r[ 2] = 12435678,   x[ 2] = 1,      y[ 2] = -9,
r[ 3] = 11435688,   x[ 3] = -10,     y[ 3] = 91,
r[ 4] = 999990,     x[ 4] = 11,      y[ 4] = -100,
r[ 5] = 435798,     x[ 5] = -131,     y[ 5] = 1191,
r[ 6] = 128394,     x[ 6] = 273,      y[ 6] = -2482,
r[ 7] = 50616,      x[ 7] = -950,     y[ 7] = 8637,
r[ 8] = 27162,      x[ 8] = 2173,     y[ 8] = -19756,
r[ 9] = 23454,      x[ 9] = -3123,    y[ 9] = 28393,
r[10] = 3708,       x[10] = 5296,     y[10] = -48149,
r[11] = 1206,       x[11] = -34899,   y[11] = 317287,
r[12] = 90,         x[12] = 109993,   y[12] = -1000010,
r[13] = 36,         x[13] = -1464808, y[13] = 13317417,
r[14] = 18,         x[14] = 3039609,  y[14] = -27634844,
```

出力 :

```
gcd(1234567890, 135792468) = 18,  x = 3039609,  y = -27634844
```

ユークリッドのアルゴリズム (メモリ節約 version)

入力 : 整数 a, b

出力 : $\gcd(a, b)$

Step 1: 変数 r_0, r_1, r_2 を宣言。

Step 2: $r_0 \leftarrow a, r_1 \leftarrow b$.

Step 3: $r_1 = 0$ ならば $\text{abs}(r_0)$ を出力。

Step 4: $r_2 \leftarrow (r_0 \% r_1), r_0 \leftarrow r_1, r_1 \leftarrow r_2$ として Step 3 へ。

拡張ユークリッドアルゴリズム (メモリ節約 version)

入力 : 整数 a, b

出力 : $\gcd(a, b)$ と $\gcd(a, b) = ax + by$ を満たす整数 x, y

Step 1: 変数 $r_0, r_1, r_2, x_0, x_1, x_2, y_0, y_1, y_2$ を宣言。

Step 2: $(r_0, x_0, y_0) \leftarrow (a, 1, 0), (r_1, x_1, y_1) \leftarrow (b, 0, 1)$.

Step 3: $r_1 = 0$ ならば Step 5 へ。

Step 4: $q \leftarrow \lfloor r_0/r_1 \rfloor$,
 $r_2 \leftarrow r_0 - q \times r_1$,
 $x_2 \leftarrow x_0 - q \times x_1$,
 $y_2 \leftarrow y_0 - q \times y_1$,
 $r_0 \leftarrow r_1$, $r_1 \leftarrow r_2$,
 $x_0 \leftarrow x_1$, $x_1 \leftarrow x_2$,
 $y_0 \leftarrow y_1$, $y_1 \leftarrow y_2$
 として Step 3 へ。

Step 5: $r_0 < 0$ ならば $(r_0, x_0, y_0) \leftarrow (-r_0, -x_0, -y_0)$.

Step 6: (r_0, x_0, y_0) を出力。

ユークリッドのアルゴリズム (再帰 version)

入力: 整数 a, b

出力: $\gcd(a, b)$

Step 1: $b = 0$ ならば $\text{abs}(a)$ を出力。

Step 2: 引数 $(b, a \% b)$ に対して再帰呼び出しを行い、その戻り値を出力。

拡張ユークリッドアルゴリズム (再帰 version)

入力: 整数 a, b

出力: $\gcd(a, b)$ と $\gcd(a, b) = ax + by$ を満たす整数 x, y

Step 1: $b = 0$ ならば

- $a \geq 0$ のとき $(a, 1, 0)$ を出力。
- $a < 0$ のとき $(-a, -1, 0)$ を出力。

Step 2: $q \leftarrow \lfloor a/b \rfloor$, $r = a - b \times q$ とおく。

Step 3: 引数 (b, r) に対して再帰呼び出しを行い、その戻り値を (e, u, v) とする。

Step 4: $d = e$, $x = v$, $y = u - q \times v$ として (d, x, y) を出力。