アルゴリズム論特論　(塩田)

— ユークリッドのアルゴリズム —

1. 数列を用いた記述

アルゴリズムを説明・証明する為には数列を用いた記述がわかり易い。

1.1. ユークリッドのアルゴリズム

Euclidean Algorithm

入力： 整数 a, b
出力： $\gcd(a, b)$

1° 数列 $\{r_n\}$ を宣言。
2° $r_0 \leftarrow a,\ r_1 \leftarrow b,\ n \leftarrow 0.$
3° $r_{n+1} = 0$ ならば $\mathrm{abs}(r_n)$ を出力。
4° $r_{n+2} \leftarrow (r_n \ \%\ r_{n+1}),\ n \leftarrow n+1$ として 3° へ。

1.2. 拡張ユークリッドアルゴリズム

Extended Euclidean Algorithm

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

1° 数列 $\{r_n\},\ \{x_n\},\ \{y_n\}$ を宣言。
2° $(r_0, x_0, y_0) \leftarrow (a, 1, 0),\ (r_1, x_1, y_1) \leftarrow (b, 0, 1),\ n \leftarrow 0.$
3° $r_{n+1} = 0$ ならば 5° へ。
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 3° へ。
5° $r_n < 0$ ならば $(r_n, x_n, y_n) \leftarrow (-r_n, -x_n, -y_n).$
6° (r_n, x_n, y_n) を出力。

1.3. アルゴリズムの根拠

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

1.4. 実行例

n	$q = \lfloor r_{n-2} \ / \ r_{n-1} \rfloor$	r_n	x_n	y_n
0		525	1	0
1		360	0	1
2	$\lfloor 525 \ / \ 360 \rfloor = 1$	165	1	-1
3	$\lfloor 360 \ / \ 165 \rfloor = 2$	30	-2	3
4	$\lfloor 165 \ / \ 30 \rfloor = 5$	15	11	-16
5	$\lfloor 30 \ / \ 15 \rfloor = 2$	0		

$\Rightarrow \gcd(525, 360) = 15 = 525 \times 11 + 360 \times (-16)$

2. メモリ節約 version

数列は直前の 2 項しか必要としないので、次の様に記述すればメモリを節約できる。

2.1. ユークリッドのアルゴリズム

Euclidean Algorithm

入力： 整数 a, b
出力： $\gcd(a, b)$

1° 変数 r_0, r_1, r_2 を宣言。
2° $r_0 \leftarrow a,\ r_1 \leftarrow b$.
3° $r_1 = 0$ ならば $\mathrm{abs}(r_0)$ を出力。
4° $r_2 \leftarrow (r_0 \ \% \ r_1)$, $r_0 \leftarrow r_1,\ r_1 \leftarrow r_2$ として 3° へ。

2.2. 拡張ユークリッドアルゴリズム

Extended Euclidean Algorithm

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

1° 変数 r_0, r_1, r_2, x_0, x_1, x_2, y_0, y_1, y_2 を宣言。
2° $(r_0, x_0, y_0) \leftarrow (a, 1, 0),\ (r_1, x_1, y_1) \leftarrow (b, 0, 1)$.
3° $r_1 = 0$ ならば 5° へ。
4° $q \leftarrow \lfloor r_0 / r_1 \rfloor$,
$(r_2, x_2, y_2) \leftarrow (r_0 - q \times r_1,\ x_0 - q \times x_1,\ y_0 - q \times y_1)$,
$(r_0, x_0, y_0) \leftarrow (r_1, x_1, y_1),\quad (r_1, x_1, y_1) \leftarrow (r_2, x_2, y_2)$
として 3° へ。
5° $r_0 < 0$ ならば $(r_0, x_0, y_0) \leftarrow (-r_0, -x_0, -y_0)$.
6° (r_0, x_0, y_0) を出力。

3. 再帰 version

推奨はできないが、再帰的に記述すると次のようになる。

3.1. ユークリッドのアルゴリズム

Euclidean Algorithm

入力： 整数 a, b
出力： $\gcd(a, b)$

1° $b = 0$ ならば $\mathrm{abs}(a)$ を出力。
2° 引数 $(b, a \,\%\, b)$ に対して再帰呼び出しを行い、その返り値を出力。

3.2. 拡張ユークリッドアルゴリズム

Extended Euclidean Algorithm

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

1° $b = 0$ ならば
　$a \geqq 0$ のとき $(a, 1, 0)$ を出力。
　$a < 0$ のとき $(-a, -1, 0)$ を出力。
2° $q \leftarrow \lfloor a/b \rfloor,\ r \leftarrow a - b \times q$ とおく。
3° 引数 (b, r) に対して再帰呼び出しを行い、その帰り値を (e, u, v) とする。
4° $(d, x, y) \leftarrow (e, v, u - q \times v)$ を出力。