アルゴリズム B 第 1 回　　2019 年 10 月 7 日

1　イントロダクション及びガイダンス

アルゴリズム

アルゴリズム (algorithm) とは，問題を解くための（明確に定められた）手順のことをいう．アルゴリズムには**入力** (input) と**出力** (output) があり，与えられた入力をもとに出力が生成される．アルゴリズムが**正当である** (correct) とは，すべての入力に対して常に停止し，その出力が正しいことをいう．

例． **ユークリッドの互除法** (Euclidean algorithm) は，最大公約数を求めるアルゴリズムであり，擬似コードを用いて次のように書ける．(a, b は非負整数とする．)

GCD(a, b)
  while $b \neq 0$
    $r = a \bmod b$
    $a = b$
    $b = r$
  return(a)

再帰呼び出しを用いて次のように書くこともできる．

GCD(a, b)
  if $b = 0$ **then**
    return(a)
  else
    return(GCD($b, a \bmod b$))

例． 自然数 n の階乗 $n!$ を求めるアルゴリズムは次のように書ける．

FACTORIAL(n)
  $r = 1$
  for $i = 1$ **to** n
    $r = ri$
  return(r)

漸近記法

アルゴリズムの計算量を評価するために，以下の漸近記法が用いられる．
関数 $f(n)$, $g(n)$ を考える．

- ある実数 $c > 0$ と非負整数 n_0 が存在して，任意の非負整数 n に対し，

$$n \geq n_0 \Longrightarrow 0 \leq f(n) \leq cg(n)$$

が成り立つとき，

$$f(n) = O\bigl(g(n)\bigr)$$

と表す．
- ある実数 $c > 0$ と非負整数 n_0 が存在して，任意の非負整数 n に対し，

$$n \geq n_0 \Longrightarrow f(n) \geq cg(n) \geq 0$$

が成り立つとき，

$$f(n) = \Omega\bigl(g(n)\bigr)$$

と表す．
- $f(n) = O\bigl(g(n)\bigr)$ かつ $f(n) = \Omega\bigl(g(n)\bigr)$ が成り立つとき，$f(n) = \Theta\bigl(g(n)\bigr)$ と表す．
- 任意の実数 $c > 0$ に対し，ある非負整数 n_0 が存在して，任意の非負整数 n に対し，

$$n \geq n_0 \Longrightarrow 0 \leq f(n) < cg(n)$$

が成り立つとき，

$$f(n) = o\bigl(g(n)\bigr)$$

と表す．
- 任意の実数 $c > 0$ に対し，ある非負整数 n_0 が存在して，任意の非負整数 n に対し，

$$n \geq n_0 \Longrightarrow f(n) > cg(n) \geq 0$$

が成り立つとき，

$$f(n) = \omega\bigl(g(n)\bigr)$$

と表す．

例． $f(n) = 2n^2 + 3n + 4$ とする．

- $f(n) = O(n^2)$ かつ $f(n) = \Omega(n)$ である．したがって，$f(n) = \Theta(n)$ である．
- $f(n) = O(n^3)$ でもある．一方，$f(n) = \Omega(n^3)$ ではない．

例． $n = o(2^n)$, $2^n = \omega(n)$ である．

注意．

- 通常の等式とは異なり，$f(n) = O(g(n))$ を $O(g(n)) = f(n)$ と書いてはならない．
- $f(n) + O(g(n))$ のように，式の中に漸近記法が現れる場合もある．この場合，$O(g(n))$ は $h(n) = O(g(n))$ を満たすある関数 $h(n)$ を表す．他の記号の場合も同様である．
- 他分野では，$O(n)$, $o(n)$ の定義において，$0 \leq f(n) \leq cg(n)$ の代わりに $|f(n)| \leq c|g(n)|$ とすることもある．