

アルゴリズムとプログラム

関連: pp.78-83

アルゴリズム 問題を解く(答を得る)ための処理手順・段取り

- 1つの問題を解くアルゴリズムは複数ある。
 - どの解き方が最も適切か?
 - トレードオフを考慮する必要がある。

「手順の明確化 & わかりやすい表現」
⇒ 目的に応じた最適なアルゴリズムを検討・決定できることが重要。

プログラミング コンピュータが処理できるようにアルゴリズムを記述すること

- あいまいさは厳禁!

「少しご飯をちょうだい」
↓
「3分の1ご飯をちょうだい」

プログラム言語を用いて、アルゴリズムをコンピュータに伝える

コンピュータに自動実行させる ⇒ **問題解決** に有効活用

フローチャート(流れ図)の記号(JIS)

関連: pp.78-79

フローチャート アルゴリズムをわかりやすく表現するための手段

	データの入力や出力		繰り返しのはじまり
	処理		繰り返しの終わり
	判断		表示
	手操作入力		書類
	プログラムの開始、終了		結合子

アルゴリズムの構造とフローチャート

関連: pp.78-79

構造化定理

- 順次構造 (記述順に実行)
- 分岐構造 (選択構造) 条件により場合分け
- 反復構造 (繰り返し構造) 条件下で繰り返し

ループの記号を使わずに繰り返しを表現した例。

機械語・アセンブリ言語...

関連: pp.82-83,98

機械語	アセンブリ言語	意味
01101001	LD A, B	ロード命令
01011010	LD B, C	ロード命令
10010110	ADD A, B	加算命令
10100101	CLR B	クリア命令

2進数の並び ↔ 1対1に対応

機械語(マシン語)

- コンピュータが理解できる唯一のプログラミング言語。数字(2進数)の並びで構成される。つまりonとoffの並びである。

アセンブリ言語

- 機械語を分かりやすくするために、簡略化した英単語(ニーモニック)で置き換えたもの(機械語を人間が書くのは大変!だから)。
- 機械語に1対1対応しており、アセンブリ言語でプログラムを書くことは機械語を書くことに等しい。

アセンブラ

- アセンブリ言語を機械語(数字の並び)に変換するプログラム。

他のプログラミング言語...

- 機械語を意識しなくてもプログラミングを行えるようになっている(機械語と1対1対応ではない)。これらは高水準言語(高級言語)と呼ばれる。

整列アルゴリズム(並び替え)

関連: p.81

交換ソート (バブルソート) 左

- 1番目とその次を比較, 2番目とその次を比較, 3番目とその次を比較... ⇒ (一番後ろが確定)
- 同様の操作を最後の1つ前まで繰り返す ⇒ (後ろから2番目が確定)
- 同様の操作を最後の2つ前まで繰り返す, 3つ前まで繰り返す, 4つ前まで繰り返す... ⇒ (後ろから3番目確定, 4番目確定...)

選択ソート 右

- 先頭と2番目を比較, 先頭と3番目を比較, 先頭と4番目... ⇒ (先頭が確定)
- 2番目と3番目を比較, 2番目と4番目を... ⇒ (2番目が確定)
- 同様の操作... ⇒ (前から3番目確定, 4番目確定...)

探索のアルゴリズム

関連: p.80

前提: 1つずつしか確認できない!

順次探索(単純前方探索)

- 探したいデータを先頭から順番に一つひとつ確認していく方法。
 - メリット: データがでたらめに並んでいても通用する。
 - デメリット: 効率に課題が残る。

二分(法)探索

- 昇順または降順に並んだデータの真ん中で二分して、探したいデータが含まれている可能性のある方を、探索範囲とする。これを繰り返す方法。
 - メリット: データが多くても、検索に要する時間が短い。
 - デメリット: データは順番に並んでいる必要がある。アルゴリズムは複雑になる。