

情報リテラシー（第7回 後期）

アルゴリズム

今日のねらい

- アルゴリズムの基本的な概念を理解できる
- フローチャートの読み方・書き方を習得できる
- 3つの整列アルゴリズム（バブル・選択・挿入）を理解できる
- 可視化ツールを使ってアルゴリズムの動きを確認できる

アルゴリズムとは

アルゴリズム (Algorithm) の定義

- 問題を解決するための**手順**のこと
- コンピュータだけでなく、日常生活にも多く存在する
- **明確で、誰が実行しても同じ結果**になるのが特徴

身近な例

- **料理のレシピ**：手順通りに作れば同じ料理ができる
- **道順の案内**：手順をたどれば目的地に到達する
- **数学の筆算**：ルールに従って計算する方法

なぜアルゴリズムが重要？

- 効率的な問題解決が可能
- 複雑な処理を整理して理解できる
- プログラミングの基礎となる

アルゴリズムの3つの基本構造

主な制御構造

1. 順次 (Sequence)

- 上から順に処理を行う
- 例：「起きる → 顔を洗う → 朝食を食べる」

2. 分岐 (Selection)

- 条件により処理を変える
- 例：「雨なら傘を持っていく / 晴れならそのまま出かける」

3. 反復 (Repetition)

- 同じ処理を繰り返す
- 例：「お皿がなくなるまで洗う」

→ この3つを組み合わせることで、ほとんどのアルゴリズムを表現できる

フローチャートとは

フローチャート（Flowchart）の定義

- アルゴリズム（手順）を **図で表現したもの**
- 処理の流れを矢印でつないで示す
- プログラムを書く前の設計にも使われる

フローチャートの利点

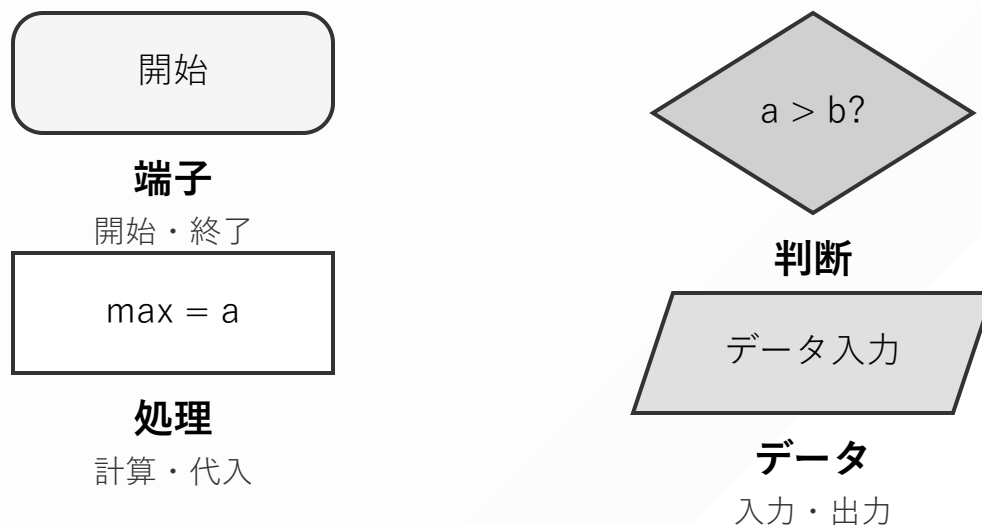
- **視覚的に理解しやすい**： 処理の流れが一目でわかる
- **共通言語**： 誰が見ても同じように理解できる
- **設計・検証に便利**： プログラムの論理ミスを発見しやすい

標準化

- 記号は **JIS X 0121** で決められている
- 世界共通の表記ルールがある

フローチャートの基本記号

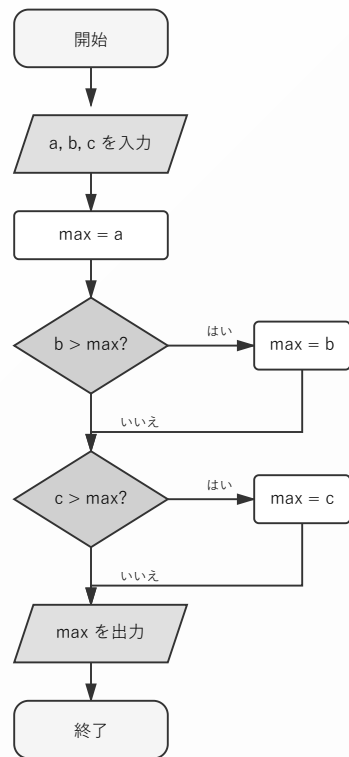
フローチャートの基本記号



矢印と流れ

- **矢印**で処理の流れを示す
- 基本は**上から下、左から右**
- 判断からは**2つの矢印**（はい／いいえ）

フローチャートの例：3つの数の中から最大値を見つける



整列（ソート）アルゴリズムとは

ソート（Sorting）の定義

- データを**一定の順序に並べ替える**処理
- 昇順（小さい順）または降順（大きい順）

身近な例

- **SNS**：タイムラインを新しい順に表示
- **ECサイト**：商品を価格順・人気順に並べる
- **成績表**：点数順にランキング

なぜソートが重要？

- データの検索が速くなる
- 情報を見やすく整理できる
- 最も基本的なアルゴリズムの1つ

整列アルゴリズム①：バブルソート

バブルソート（Bubble Sort）の仕組み

- **隣り合う要素を比較**して入れ替える
- 大きい値が泡のように右へ移動する

手順

1. 隣り合う2つの値を比較する
2. 左が右より大きければ入れ替える
3. 左端から右端まで繰り返す
4. 一番大きな値が一番右に移動する
5. これを必要な回数だけ繰り返す

特徴

- **利点**：考え方も実装もシンプル
- **欠点**：比較と入れ替えがとても多い → 大きなデータには不向き

バブルソートの例

データ：[5, 2, 8, 1, 9]

1回目の走査

```
[5, 2, 8, 1, 9] → 5と2を比較して入れ替え  
[2, 5, 8, 1, 9] → 5と8はそのまま  
[2, 5, 8, 1, 9] → 8と1を比較して入れ替え  
[2, 5, 1, 8, 9] → 8と9はそのまま  
[2, 5, 1, 8, 9] → 最大値9が右端に確定
```

2回目の走査

```
[2, 5, 1, 8, 9] → 2と5はそのまま  
[2, 5, 1, 8, 9] → 5と1を比較して入れ替え  
[2, 1, 5, 8, 9] → 5と8はそのまま  
[2, 1, 5, 8, 9] → 8が右から2番目に確定
```

→ これを繰り返して完成

整列アルゴリズム②：選択ソート

選択ソート（Selection Sort）の仕組み

- 未整列部分から**最小値を選択**する
- 選んだ最小値を先頭に移動

手順

1. 未整列の部分から**最小値**を探す
2. 見つけた最小値を、未整列部分の一番左と入れ替える
3. 「整列済みの範囲」を1つ右に広げる
4. 同じことを繰り返す

特徴

- **利点**： 最小値を探して先頭に持ってくる単純な動き
- **性質**： 比較回数は多いが、入れ替えは少なめ
- バブルソートよりは少し合理的

選択ソートの例

データ：[5, 2, 8, 1, 9]

1回目

[5, 2, 8, 1, 9] → 最小値1を探す
[1, 2, 8, 5, 9] → 1と5を入れ替え（1が確定）

2回目

[1, 2, 8, 5, 9] → 残りの最小値2を探す
[1, 2, 8, 5, 9] → すでに正しい位置（2が確定）

3回目

[1, 2, 8, 5, 9] → 残りの最小値5を探す
[1, 2, 5, 8, 9] → 5と8を入れ替え（5が確定）

→ 残りも確定して完成

整列アルゴリズム③：挿入ソート

挿入ソート（Insertion Sort）の仕組み

- 整列済み部分に新しい値を**挿入**する
- トランプの手札を整理する方法に似ている

手順

1. 左側を「すでに整列済み」とみなす
2. 新しい値を1つ取り出す
3. 左側の整列済み部分の中で、入るべき場所を探す
4. 必要な要素を右にずらして、すき間に「挿入」する

特徴

- **利点**： ほぼ並んでいるデータに強い（少しの入れ替えで済む）
- **適用**： 小さいデータならとても速い
- 人がトランプを並べ替えるときのやり方に近い

挿入ソートの例

データ：[5, 2, 8, 1, 9]

1回目（2を挿入）

[5] | [2, 8, 1, 9] → 5は整列済み、2を挿入
[2, 5] | [8, 1, 9] → 2を5の前に挿入

2回目（8を挿入）

[2, 5] | [8, 1, 9] → 8は5より大きいのでそのまま
[2, 5, 8] | [1, 9]

3回目（1を挿入）

[2, 5, 8] | [1, 9] → 1を先頭に挿入
[1, 2, 5, 8] | [9]

→ 最後に9を挿入して完成

3つのソートの効率比較

比較回数の目安（最悪の場合）

アルゴリズム	データ10件	データ100件	データ1000件	特徴
バブルソート	約45回	約5,000回	約500,000回	比較と入れ替えがとても多い
選択ソート	約45回	約5,000回	約500,000回	最小値を探すので比較は多い
挿入ソート	約45回	約5,000回	約500,000回	ほぼ整列済みなら大幅に速い

ポイント

- データ量が増えるほど、アルゴリズムの「効率」の差が大きくなる
- **挿入ソート**はデータがほぼ整列済みの場合に特に高速
- 実用では、データの特성에応じてアルゴリズムを選ぶ

VisuAlgo：アルゴリズム可視化ツール

VisuAlgoとは

- アルゴリズムの動きを**視覚的に確認**できるWebツール
- ソート、探索、グラフなど様々なアルゴリズムに対応
- **無料**で誰でも使える

アクセス方法

- **URL**： <https://visualgo.net/>
- ブラウザで開くだけ（インストール不要）

準備

1. サイトを開く
2. 右上の言語設定で「**日本語**」を選択
3. メニューから「**Sorting**」を選ぶ

演習：VisuAlgoでソートを観察しよう

手順

1. VisuAlgoで「Sorting」を開く
2. 以下の3つのアルゴリズムをそれぞれ実行
 - **Bubble Sort**（バブルソート）
 - **Selection Sort**（選択ソート）
 - **Insertion Sort**（挿入ソート）
3. 「Go」ボタンで実行して、棒グラフの動きを観察

演習：データパターンを変えて比較

データパターンの変更

画面上部のボタンで以下のパターンを試してみよう

パターン1：ランダム (Random)

- ばらばらに並んだデータ

パターン2：整列済み (Sorted)

- すでに正しく並んでいるデータ

パターン3：逆順 (Reversed)

- 大きい順に並んだデータ

観察ポイント

- どのアルゴリズムが一番よく動き回っているか？
- ほぼ整列済みのときに速いのはどれか？

身近なアルゴリズムの活用例

Webサービスでの並べ替え

- **SNS**：タイムラインを「新しい順」に並べる
- **ECサイト**：商品を「価格の安い順／高い順」に並べる
- **動画サイト**：再生回数順・人気順に表示
- **学校**：成績表を点数順にソート

検索エンジン

- 検索結果を関連度順に並べる
- ページランクアルゴリズムで重要度を計算

身近な機器

- 音楽プレーヤーの曲順並べ替え
- 写真アプリの日付順・名前順表示

アルゴリズムのポイント

重要な考え方

- 「やり方（手順）」が決まっているから、**毎回同じ結果**になる
- データ量が増えると、**効率の良いアルゴリズム**が重要になる
- 問題に応じて**最適なアルゴリズムを選ぶ**ことが大切

アルゴリズムの評価

- **正しさ**：正しい答えを出せるか
- **速さ**：どれくらい時間がかかるか
- **わかりやすさ**：理解しやすく実装しやすいか

→ 実用では、これらのバランスを考えて選択する

まとめ

- **アルゴリズム**は問題を解くための手順
- **フローチャート**で処理の流れを図として表せる
- **3つの基本構造**（順次・分岐・反復）でアルゴリズムを表現
- **バブルソート・選択ソート・挿入ソート**にはそれぞれ特徴がある
 - バブル：シンプルだが効率は低い
 - 選択：最小値を選んで並べる
 - 挿入：ほぼ整列済みデータに強い
- **VisuAlgo**で実際の動きを見ると、違いが分かりやすい
- アルゴリズムは現代社会のあらゆる場面で使われている

今日の感想や気づきをTeamsのチャット欄に書いてください。