

ソートアルゴリズム

プログラミング7th

ソート（整列）アルゴリズムとは、

バラバラに並んだデータを一定の規則（昇順や降順）に従って
並べ替える手順のこと

「選択ソート」や「バブルソート」、「挿入ソート」は
教育用として優れている

実務や大量のデータ処理では、より高度なアルゴリズムが使われている

□高速なアルゴリズム

現代のプログラミング言語の標準機能でよく使われる、非常に効率の良い手法

○クイックソート (Quick Sort)

「ピボット (軸)」となる値を一つ決め、それより小さいグループと大きいグループに分割していく作業を繰り返す手法

特徴：平均的に最も高速とされ、多くのライブラリで採用

計算量： $O(n \log n)$

○マージソート (Merge Sort)

データを最小単位まで分割してから、順番を守りつつ合体 (マージ) させていく手法

特徴：安定ソートであり、データがどんな状態でも必ず一定の速度で終わる

計算量： $O(n \log n)$

□ どう使い分けるのか？

実は、どれか一つが最強というわけではなく、状況に応じて「ハイブリッド」で使われることが多い

○ 基本はクイックソートやマージソート

大半のケースでこれらが使われる

○ データが少ないなら挿入ソート

実はデータ数が10~20個程度なら、複雑なクイックソートよりもシンプルな挿入ソートの方が速いことが多いため、クイックソートの内部で「最後は挿入ソートに任せる」という処理がよく行われる

○ 安定性が大事ならマージソート

元の順序（出席番号など）を壊したくない場合は、クイックソートではなくマージソートを選ぶ

並べ替えアルゴリズムの使用場面・活用例

分野	具体的な活用例	アルゴリズムの選択理由
日常生活	電話帳、辞書、図書館の蔵書	アルファベット順に並べることで、目的の情報を迅速に検索するため
コンピュータサイエンス	データベース、検索エンジン、グラフ描画、機械学習、データ分析	データを特定の順序に並べることで、効率的な処理や分析が可能になるため
プログラミング	配列のソート、カスタムデータ構造のソート、アルゴリズムの補助	プログラムの処理を効率化し、正確な結果を得るため

本時の目標

ソートアルゴリズムの概念を理解し、
それぞれの特徴をもとに比較すること。

- 協力・共同して取り組もう
- アルゴリズムを理解しよう
- コンピュータの仕組み（並び替え）を読み取ろう

アジェンダ（本時の学習の流れ）

1. はじめに（目標の確認）
2. エキスパート
3. ジグソー
4. (クロストーク)
5. リフレクション

エキスパート活動

課題 A 「選択ソート」

課題 B 「バブルソート」

課題 C 「選択ソート vs バブルソート」

○資料を読んで話し合いながら理解を深めていこう。

○専門家になろう。説明できることを目標に取り組もう。

<環境> Google Colaboratory

【Aエキスパ】 プロ7th 『選択ソートプログラム』 .ipynb

【Bエキスパ】 プロ7th 『バブルソートプログラム』 .ipynb

【Cエキスパ】 プロ7th 『選択ソート×バブルソート』 .ipynb

ジグソー活動

○エキスパート活動の共有

- ・まずは資料を説明しよう！

○ジグソー課題(1)『拳動のシミュレーション』

以下の数値が並んだリストを、昇順（小さい順）に並べ替える場面を想像してください。対象データ：[5, 3, 8, 2]

①選択ソートを使用した場合、「1回目の最小値探索と交換」が終わった直後のリストの状態を教えてください。

②バブルソートを使用した場合、「隣り合う要素の比較・交換」が1周（1パス）終わった直後のリストの状態を教えてください。

ジグソー活動

○ジグソー課題(2) 『特性の理解』

次のような特性を持つデータセットに対して、選択ソートとバブルソートのどちらが「より効率的」か、理由とともに答えてください。

ケースA：すでにほとんど昇順に並んでいるが、一部だけ順序が入れ替わっているデータ。

ケースB：データの入れ替え（スワップ）コストが非常に高く、できるだけ交換回数を抑えたいシステム。

ジグソー活動

○ジグソー課題(3) 『安定性の考察』

「安定なソート (Stable Sort)」 についての問題です。

データ： [(A, 80), (B, 50), (C, 80)]

※ (名前, スコア) の形式

このデータを「スコアの低い順」に並べ替えます。バブルソートの結果として正しいものはどちらですか？また、選択ソートで並べ替えたとき、意図せず「AとCの順序」が入れ替わってしまう可能性があるのはなぜですか？

候補①： [(B, 50), (A, 80), (C, 80)]

候補②： [(B, 50), (C, 80), (A, 80)]

クロストーク

○他の班の考えを聞いてみよう

- ・ 3人のうち1人はその場に残り他者に説明をする。
- ・ 3人のうち2人は他のグループに移動し、情報を収集する。
- ・ 代表者が説明した後は、問答をする。

※多様な意見を聞きながら、深い理解へと繋げよう

代表的な並び替えアルゴリズムの比較

アルゴリズム	特徴	適している状況
バブルソート	シンプルだが、データ量が多いと非効率	小規模なデータや、安定性を重視する場合
選択ソート	安定性があり、メモリ使用量が少ない	小規模なデータや、メモリ制限がある場合
挿入ソート	部分的にソートされたデータに対して効率的	小規模なデータや、ほぼソート済みのデータの場合
クイックソート	平均的な性能が非常に高く、広く利用されている、 $O(n \log n)$ の時間計算量	大規模なデータに対して高速
マージソート	安定性があり、最悪の場合でも $O(n \log n)$ の時間計算量	大規模なデータで安定性を重視する場合
ヒープソート	空間計算量が $O(n)$ で高速	大規模なデータで、インプレースなソートが必要な場合

リフレクション

- ソートアルゴリズムの概要を理解した。
- ソートアルゴリズムの活用事例を知った。
- 選択ソートとバブルソートのアルゴリズムを理解し、それらを比較した。