

# ソートの基本

## 1 導入

この講義で最重要なのは、ソートを「小さい順に並べる作業」でなく、順序の情報を作る操作として見ることです。

探索や二分探索が効率的に働くのは、ソートで順序が作られているからです。この講義では、ソート自体の細かい実装より、何を比較し、どう並べ替えているかを整理します。

## 2 用語と定義

ソートとは、要素列のある規則にしたがって整列することです。

## 3 方針

まず比較を何回するかを見ます。そのあと、要素を入れ替えるのか、分けてから戻すのかという動かし方を見ます。

## 4 直感的な説明

ソートは、散らばった本を本棚の順番に並べ直すようなものです。近くの2冊を少しずつ入れ替える方法もあれば、大きく2組に分けてから整える方法もあります。

## 5 厳密な説明

### 5.1 1. 基本の比較

要素が  $n$  個あるとき、単純な比較ベースのソートでは  $O(n^2)$  の回数だけ比較するものが多いです。

### 5.2 2. 分割統治

マージソートのように、列を半分へ分けてそれぞれをソートし、最後に併合する方法では  $O(n \log n)$  が目安になります。

### 5.3 3. ソートで得られるもの

ソート済みになると、最小値・最大値・重複の確認・二分探索など、多くの処理が簡単になります。

## 6 見分け方

- 順序があると後続の処理が楽になるなら、まずソートを疑います。
- 入力 が大きいなら  $O(n^2)$  か  $O(n \log n)$  かを意識します。

## 7 最終形

sort = 順序情報を作る前処理

## 8 一言でいうと

- ソートは並べ替え作業そのものより、その後の処理を簡単にするための基盤です。