第3章　ソート（並び替え）

大小関係が定められた複数の要素からなるデータの集合を一定の規則に従って並べ替えること

第1節 はじめに

効率的なソートとは、ソート済みされたデータを必要とするとき、他のアルゴリズム（探索やマージ）でこのソート済みのデータを使用したり、あるいは処理したしする時、アルゴリズムが最適化であるときのデータでもある

1.1 ソートアルゴリズム

計算機科学では、ソートアルゴリズムを次のように分類

- ◆ **安定ソート(stable sort)**：並び替えのアルゴリズムのうち、同等なデータのソート前の順序が、ソート後も保存されるものをいう。つまり、ソート途中の各状態において、常に順位の位置関係を保っていることをいう。
- ◆ **内部ソートと外部ソート**：ソートされるデータの格納領域を変更して処理を進めていくIn-place のソートを内部ソートという。一方、ソートされるデータの格納領域以外にO(n) 以上の一時的な記憶領域が必要であるソートを外部ソートという。
- ◆比較ソート：個々の項目を比較演算で大小判定することを基本とするソートを比較ソートという。計算理論において、n 個のデータのソートは、データの大小比較のみによって行う場合、最悪計算量が最低でもO(n log n) 必要なことが知られている。

1.2 計算量

並び替えるデータの項目数 n に基づいた計算量は、典型的なソートアルゴリズムでは、最善で O(n log n) 、最悪で O(n^2) である。理想は O(n) である。比較ソートでは、必ず O(n log n) の比較が必要となる。

並び替えるデータの項目数 n に基づいた計算量は、典型的なソートアルゴリズムでは、最善で O(n log n) 、最悪で O(n^2) である。理想は O(n) である。比較ソートでは、必ず O(n log n) の比較が必要となる。

Theorem 1.1

計算理論において、n 個のデータのソートは、データの大小比較のみによって行う場合、最悪計算量が最低でも $\mathrm{O}(n \log n)$ 必要である

[証明]

n 個のデータの並べ替えは置換の数と、判定 (if 文) の数で決めれる。

置換は最悪でも $n!$ 回である。スターリングの公式 (ガンマ関数 (或いは階乗) の漸近近似) より、$n! = \mathrm{O}(n^n)$ で表される。

一方、判定 (if 文) での分岐は、1 回の if 文で 2 つに分岐するので、最終的に if 文の分岐の数を m 回とすると、次式で無ければ成らない

$$2^m \geq n! = \mathrm{O}(n^n)$$

上式を対数 (log) をとると、次式が得られ、最悪の計算量は最低でも次式となる

$$m = \mathrm{O}(n \log n)$$

■

1.3 ソート性能

配列に格納された n 個のデータをソートする場合について、各アルゴリズムの性能を示し、計算時間の表記に 用いている記号 (オーダー) O については、ランダウの記号を用いる。

名称	平均計算時間	最悪計算時間	メモリ	安定性	手法
バブルソート	-	$O(n^2)$	O(1)	○	交換
シェーカーソート	-	$O(n^2)$	O(1)	○	交換
コムソート	$O(n \log n)$	$O(n^2)$	O(1)	×	交換
ノームソート	-	$O(n^2)$	O(1)	○	交換
選択ソート	$O(n^2)$	$O(n^2)$	O(1)	×	選択
挿入ソート	$O(n + d)$	$O(n^2)$	O(1)	○	挿入
シェルソート	-	$O(n \log^2 n)$	O(1)	×	挿入
2 分木ソート	$O(n \log n)$	$O(n \log n)$	O(n)	○	挿入
ライブラリーソート	$O(n \log n)$	$O(n^2)$	O(n)	○	挿入
マージソート	$O(n \log n)$	$O(n \log n)$	O(n)	○	マージ
In-place マージソート	$O(n \log n)$	$O(n \log n)$	O(1)	○	マージ
ヒープソート	$O(n \log n)$	$O(n \log n)$	O(1)	×	選択
スムースソート	-	$O(n \log n)$	O(1)	×	選択
クイックソート	$O(n \log n)$	$O(n^2)$	$O(\log n)$	×	パーティショニング
イントロソート	$O(n \log n)$	$O(n \log n)$	$O(\log n)$	×	混成
ペイシェンスソート	-	$O(n^2)$	O(n)	×	挿入
ストランドソート	$O(n \log n)$	$O(n^2)$	O(n)	○	選択
奇偶転置ソート	-	$O(n^2)$	O(1)	○	交換
シェアソート	-	$O(n^{1.5})$	O(1)	×	交換

比較ソート以外のソートアルゴリズム

名称	平均計算時間	最悪計算時間	メモリ	安定性	$n \ll 2^k$
鳩の巣ソート	$\mathrm{O}(n+2^k)$	$\mathrm{O}(n+2^k)$	$\mathrm{O}(2^k)$	○	○
バケットソート	$\mathrm{O}(nk)$	$\mathrm{O}(n^2k)$	$\mathrm{O}(nk)$	○	×
分布数えソート	$\mathrm{O}(n+2^k)$	$\mathrm{O}(n+2^k)$	$\mathrm{O}(n+2^k)$	○	○
LSD 基数ソート	$\mathrm{O}(nk/s)$	$\mathrm{O}(nk/s)$	$\mathrm{O}(n)$	○	×
MSD 基数ソート	$\mathrm{O}(nk/s)$	$\mathrm{O}(n(k/s2^s)$	$\mathrm{O}((k/s2^s)$	×	×
スプレッドソート	$\mathrm{O}(nk/\log(n))$	$\mathrm{O}(n(k-\log(n))^{0.5})$	$\mathrm{O}(n)$	×	×
逆写像ソート	$\mathrm{O}(n)$?	N/A	$\mathrm{O}(n)$?	○	×

ただし、k はキーの長さ、s は実装で使われるチャンクのサイズである

5 安定性

ソートアルゴリズムには安定ソートと不安定ソートがある。安定ソート (stable sort) とは、ソート (並び替え) のアルゴリズムのうち、同等なデータのソート前の順序が、ソート後も保存されるものをいう。つまり、ソート 途中の各状態において、常に順位の位置関係を保っていることをいう

Definition 5.1 (安定性)

数値が等しいデータについて、整列前の順序性がそのまま整列後に保存される

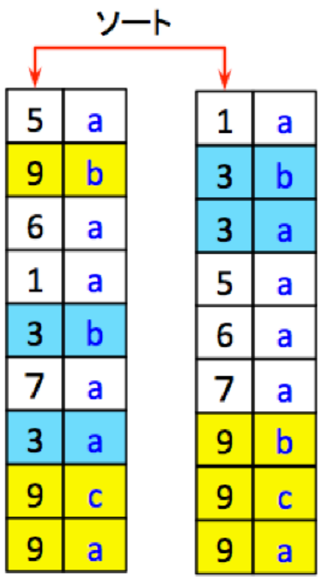

図 5.1　安定ソート

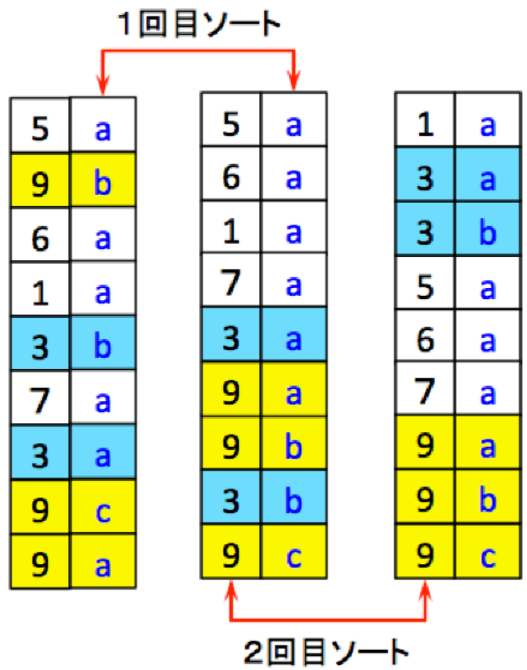

図 5.2　２項目の安定ソート