

## 7. ソート (3)

$n$  個の要素をソートするとき、ヒープソートやマージソートでは、最悪の場合の計算量が  $O(n \log n)$  である。これらのソートのように、これまで取り扱ったソートアルゴリズムはすべて 2 個の要素の比較に基づいている。このような比較に基づくソートアルゴリズムでは、どのように工夫しても最悪の場合の計算量を  $o(n \log n)$  にできないことが知られている。しかし、比較以外の方法を用いれば最悪の場合の計算量をより小さくすることが可能である。

- ソートアルゴリズムが安定であるとは、キー（ソートに用いる情報）が等しい要素について、ソート前の順序を保つことをいう。
- バケットソート (bucket sort) またはビンソート (bin sort) は、キーの種類が限られている場合に、その種類の数だけのキュー（待ち行列）を用意し、各要素を対応するキューに入れ、順に取り出すことでソートするアルゴリズムである。
- 計数ソート (counting sort) は、キーの種類が限られている場合に、各要素の出現回数を数えることでソートするアルゴリズムである。

例えば、集合  $\{0, 1, \dots, m-1\}$  の要素からなる列の整列を考える。まず、要素  $i$  が列に現れる回数  $c[i]$  を数える。このとき、要素  $i$  はソートされた列の  $\sum_{j=0}^{i-1} c[j] + 1$  番目から  $\sum_{j=0}^i c[j]$  番目に現れる。この情報によって、元の列をソートする。

- 基数ソート (radix sort) は、バケットソートや計数ソートなどの高速で安定なソートを繰り返し適用することでソートするアルゴリズムである。

例えば、10 進法で表された 3 桁の正の整数の列を考える。まず一の位についてバケットソートや計数ソートなどでソートする。次に十の位、最後に百の位について同様にソートする。これによって全体のソートが完了する。

より一般に、 $b$  を 2 以上の整数とする。正の整数からなる列について、各要素を  $b$  進法で表し、最下位から最上位まで順に各桁をキーとしてバケットソートや計数ソートなどを適用することで列全体がソートされる。

データの種類の数が要素数  $n$  に比べて小さいとき、上で述べたアルゴリズムはどれも計算量が  $O(n)$  となる。ただし、ソートできるデータの種類の制限があることに注意せよ。また、上で述べたアルゴリズムは（適切に実装すれば）どれも安定である。

## 問題

- 7-1. 前ページの基数ソートが整数列を正しくソートすることを示せ. ただし, 基数ソートに用いるバケットソートや計数ソートが安定であることは証明せずに用いてよい.

以下の2問では, 集合  $S = \{a, b, c, d\} \times \{1, 2, 3, 4\}$  の要素の列

$$(c, 4), (c, 2), (b, 3), (d, 1), (c, 3), (a, 1) \quad (*)$$

を考える.

- 7-2. 列 (\*) を第1成分のみをキーとしてバケットソートによって昇順にソートし, その経過を図示せよ.

- 7-3. 列 (\*) を第2成分のみをキーとしてバケットソートによって昇順にソートし, その経過を図示せよ.

- 7-4. 次の1以上4以下の整数からなる列を計数ソートによって昇順にソートし, その経過を図示せよ.

$$4, 4, 2, 2, 3, 2, 1, 2, 2, 3$$

- 7-5. 次の1以上9以下の整数からなる列を計数ソートによって昇順にソートし, その経過を図示せよ.

$$9, 3, 3, 7, 6, 1, 3, 8, 2, 1$$

- 7-6. 次の整数列を基数ソートによって昇順にソートし, その経過を図示せよ. (バケットソートや計数ソートの経過は省略してよい. すなわち, 一の位, 十の位, 百の位でソートした結果をそれぞれ書けばよい.)

$$442, 735, 194, 367, 702, 652, 805, 726, 456, 867$$

- 7-7. 前問と同様に, 次の整数列を基数ソートでソートする経過を図示せよ.

$$184, 446, 928, 180, 363, 758, 505, 366, 290, 525$$