

5. ソート (1)

要素の列を大きさの順に並び替えることをソート（整列）という。

ソートのアルゴリズムには様々なものが知られているが、今回は以下のものを取り扱う。以下では、列 $a[0], a[1], \dots, a[n-1]$ を昇順に並び替えるものとして説明する。

バブルソート 列 $a[0], a[1], \dots, a[n-1]$ を先頭から見ていき、隣接している要素の順序が逆転していたら交換する。これを最後まで行くと、 $a[n-1]$ が最大値となるので、列 $a[0], a[1], \dots, a[n-2]$ に同様の操作を行う。以下同様に繰り返す。アルゴリズムは次のように書ける。

```
for  $i \leftarrow n-1, n-2, \dots, 1$  do
  for  $j \leftarrow 1, 2, \dots, i$  do
    if  $a[j-1] > a[j]$  then
       $a[j-1]$  と  $a[j]$  を交換する
    end if
  end for
end for
```

選択ソート 列 $a[0], a[1], \dots, a[n-1]$ から最小の要素を選び、それを $a[0]$ と交換する。次に、列 $a[1], a[2], \dots, a[n-1]$ から最小の要素を選び、それを $a[1]$ と交換する。以下同様に繰り返す。アルゴリズムは次のように書ける。

```
for  $i \leftarrow 0, 1, \dots, n-2$  do
   $min \leftarrow i$ 
  for  $j \leftarrow i+1, i+2, \dots, n-1$  do
    if  $a[min] > a[j]$  then
       $min \leftarrow j$ 
    end if
  end for
   $a[i]$  と  $a[min]$  を交換する
end for
```

挿入ソート $a[0] > a[1]$ ならば $a[0]$ と $a[1]$ を交換する. このとき列 $a[0], a[1]$ はソートされている. 次に, $a[2]$ を列 $a[0], a[1]$ の中で正しい位置に挿入する. このとき列 $a[0], a[1], a[2]$ はソートされている. 以下同様に繰り返す. アルゴリズムは次のように書ける.

```
for  $i \leftarrow 1, 2, \dots, n-1$  do
   $t \leftarrow a[i]$ 
   $j \leftarrow i-1$ 
  while  $j \geq 0$  and  $a[j] > t$  do
     $a[j+1] \leftarrow a[j]$ 
     $j \leftarrow j-1$ 
  end while
   $a[j+1] \leftarrow t$ 
end for
```

問題

5-1. 次の整数列 (*) をバブルソートによって昇順にソートし, その経過を図示せよ.

98, 56, 31, 50, 29, 97, 55, 54 (*)

5-2. 整数列 (*) を選択ソートによって昇順にソートし, その経過を図示せよ.

5-3. 整数列 (*) を挿入ソートによって昇順にソートし, その経過を図示せよ.

以下の問題において, 比較回数, 交換回数は列の要素のものだけを数え, 添字の比較回数は数えないものとする.

5-4. n 個の要素の列をバブルソートによってソートするとき, 最悪の場合の比較回数, 交換回数はともに $O(n^2)$ であることを示せ.

5-5. n 個の要素の列を選択ソートによってソートするとき, 最悪の場合の比較回数は $O(n^2)$, 交換回数は $O(n)$ であることを示せ.

5-6. n 個の要素の列を挿入ソートによってソートするとき, 最良の場合の比較回数は $O(n)$, 最悪の場合の比較回数は $O(n^2)$ であることを示せ.

5-7. 前ページのバブルソートのアルゴリズムにおいて, 内側の for ループにおいて交換が起きなかったとき, 列がすでにソートされていることを示せ. (このことを使ってアルゴリズムを改良すると, 最良の場合の比較回数を $O(n)$ にすることができる.)