

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 $a[1], a[2], \dots, a[n]$ 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 $a[1], a[2], \dots, a[n]$ 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 , $a[n]$ が最大値となるので , 列 $a[1], a[2], \dots, a[n-1]$ に同様の操作を行う . 以下同様に繰り返す .

選択ソート 列 $a[1], a[2], \dots, a[n]$ から最小の要素を選び , それを $a[1]$ と交換する . 次に , 列 $a[2], a[3], \dots, a[n]$ から最小の要素を選び , それを $a[2]$ と交換する . 以下同様に繰り返す .

挿入ソート $a[1] > a[2]$ ならば $a[1]$ と $a[2]$ を交換する . 次に , $a[3]$ を列 $a[1], a[2]$ の中で正しい位置に挿入する . 以下同様に繰り返す .

シェルソート *¹ 狭義単調減少な自然数の列 h_1, h_2, \dots, h_k を固定する . ただし , $h_k = 1$ とする . アルゴリズムは次のように書ける .

```
for  $i \leftarrow 1, 2, \dots, k$  do
  for  $j \leftarrow 1, 2, \dots, h_i$  do
    部分列  $a[j], a[j + h_i], a[j + 2h_i], \dots$  を挿入ソートでソートする
  end for
end for
```

数列 $\{h_i\}$ としては , 例えば , $h_k = 1, h_{i-1} = 3h_i + 1$ で定まるものが使われる . このとき , 比較回数は最悪で $O(n^{3/2})$ であることが知られている .

ヒープソート まず , 要素の列から最大値が先頭になるヒープを作る *² . 次に , ヒープから最大値を取り出し , 最後の要素と交換する操作を繰り返すことでソートする .

*¹ シェル (Shell) は人名である .

*² プリント No. 3 では最小値を先頭にしていただけに注意せよ .

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.

87, 52, 96, 40, 99, 60, 23, 51 (*)

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.

5-4. 整数列 (*) をシェルソートによって昇順にソートし, その経過を図示せよ. ただし, 数列 $\{h_i\}$ として, $h_1 = 4, h_2 = 1$ を用いるとする.

5-5. 整数列 (*) をシェルソートによって昇順にソートし, その経過を図示せよ. ただし, 数列 $\{h_i\}$ として, $h_1 = 7, h_2 = 3, h_3 = 1$ を用いるとする.

5-6. 整数列 (*) をヒープソートによって昇順にソートし, その経過を図示せよ.

5-7. 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, 最悪時の比較回数は $O(n^2)$ であることを示せ.

5-8. 整数列 $a[1], a[2], \dots, a[n]$ を選択ソートによってソートするアルゴリズムを書け. (このプリントにあるような擬似コードでも, C 言語等のプログラミング言語でもよい.)

5-9. n 個の要素の列をヒープソートによってソートするとき, 最悪時の比較回数は $O(n \log n)$ であることを示せ.