

ソート

ソートとは

- 表の形のデータを一定の基準に従って並べかえることをソート(sort, ソーティング、整列、並べ替え、順序付け)という
- 例えば、表1の元のデータがあるとするば、表2～表5はそれぞれ基準(目的)の違うソートとなる

表1 元のデータ

名前	身長(cm)	体重(kg)
Yasutaka	120	37
Hiroaki	123	30
Masatoshi	118	45
Hironobu	110	28
Ryosuke	125	42

基準の違うソート

表2 名前のアルファベット順

名前	身長(cm)	体重(kg)
Hiroaki	123	30
Hironobu	110	28
Masatoshi	118	45
Ryosuke	125	42
Yasutaka	120	37

表3 身長の高順

名前	身長(cm)	体重(kg)
Hironobu	110	28
Masatoshi	118	45
Yasutaka	120	37
Hiroaki	123	30
Ryosuke	125	42

基準の違うソート

表4 体重の昇順

名前	身長(cm)	体重(kg)
Hironobu	110	28
Hiroaki	123	30
Yasutaka	120	37
Ryosuke	125	42
Masatoshi	118	45

表5 体重の降順

名前	身長(cm)	体重(kg)
Masatoshi	118	45
Ryosuke	125	42
Yasutaka	120	37
Hiroaki	123	30
Hironobu	110	28

用語

- レコード: 表中の個々のデータ
- フィールド(項目): レコードを構成する名前や身長
- キー: ソートするときに基準となる(着目する)項目
 - 表2, 3, 4はそれぞれ名前, 身長, 体重がキー

選択ソート

種々のソート法の紹介

考え方

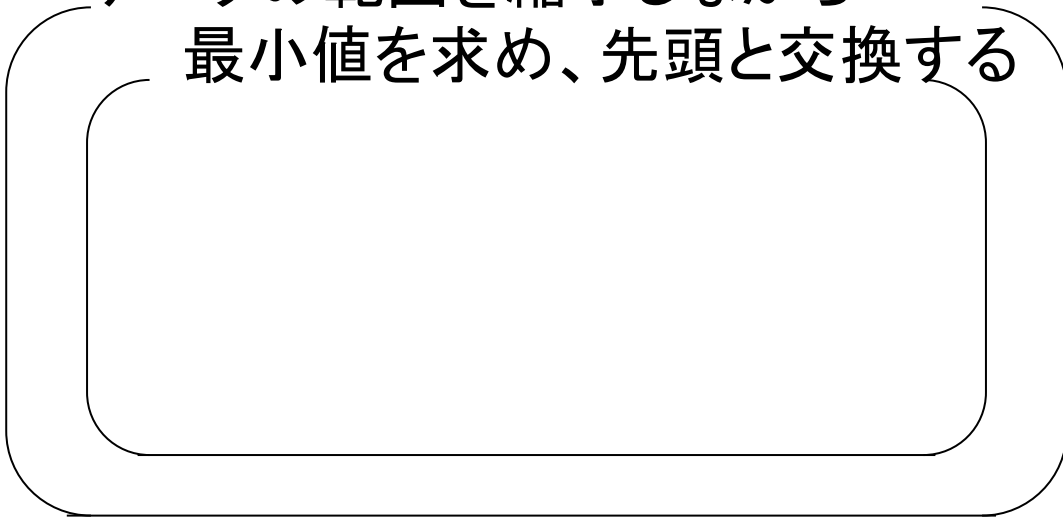
データ全体の中から最小のものを見つけ出す。次に残りの中から最小のものを見つけ出す。これらを繰り返して整列を行う

考え方(より詳細に)

- 配列全体の中から最小のものを見つけ出し、それを配列の先頭と交換
- 次に残りの中から最小のものを見つけ出し、配列の2番目と交換
- 以上のことを繰り返していく

ソート:「最小値をもとめ、先頭と交換する」 の繰り返し

データの範囲を縮小しながら
最小値を求め、先頭と交換する

The diagram consists of two nested rounded rectangles. The outer rectangle is larger, and the inner rectangle is smaller and centered within the outer one. This visualizes the process of selection sort where the range of elements being considered for the minimum value decreases with each iteration.

ソート例: (40, 30, 10, 20)

- データ全体に対し、**最小値**を選び、先頭と交換

40	30	10	20
----	----	----	----



10	30	40	20
----	----	----	----

- 先頭を除いてのデータに対し、**最小値**を選び、先頭と交換

10	30	40	20
----	----	----	----



10	20	40	30
----	----	----	----

- 先頭を除いてのデータに対し、**最小値**を選び、先頭と交換

10	20	40	30
----	----	----	----



10	20	30	40
----	----	----	----

ソートの手順

配列: $a[0], a[1], \dots, a[n-1]$

最小値: min, 位置: no

Step 0 最小値を求める

- $\text{min} = a[0]$, $\text{no} = 0$ とおく
- min と $a[j]$ ($j = 1, \dots, n-1$) を比較。 $\text{min} > a[j]$ なら $\text{min} = a[j]$, $\text{no} = j$
- $\text{no} \neq 0$ なら $a[0]$ と $a[\text{no}]$ を交換

Step 1 最小値を求める

- $\text{min} = a[1]$, $\text{no} = 1$ とおく
- min と $a[j]$ ($j = 2, \dots, n-1$) を比較。 $\text{min} > a[j]$ なら $\text{min} = a[j]$, $\text{no} = j$
- $\text{no} \neq 1$ なら $a[1]$ と $a[\text{no}]$ を交換

⋮

Step $n-2$ 最小値を求める

- $\text{min} = a[?]$, $\text{no} = ?$ とおく
- min と $a[j]$ ($j = ?, \dots, n-1$) を比較。 $\text{min} > a[j]$ なら $\text{min} = a[j]$, $\text{no} = j$
- $\text{no} \neq ?$ なら $a[?]$ と $a[\text{no}]$ を交換

選択ソートアルゴリズム

入力: 配列 $a[0], \dots, a[n-1]$

出力: ソート済みの配列 $a[0], \dots, a[n-1]$

補助: min: 最小値, no: 位置, i, j

1. $i=?$ から $??$ まで、次の処理を繰り返す。

1.1 {初期設定} $min=?$, $no=?$ とおく。

1.2 {比較} $j=?$ から $??$ まで、次の処理を繰り返す

1.2.1 もし $a[j] > min$ であれば、 $min=?$, $no=?$ とおく。

1.3 {交換} もし $no \neq i$ であれば $a[i]$ と $a[no]$ を交換

選択ソートアルゴリズム

入力: 配列 $a[0], \dots, a[n-1]$

出力: ソート済みの配列 $a[0], \dots, a[n-1]$

補助: ~~min: 最小値~~, no: 位置, i, j

1. $i=?$ から $??$ まで、次の処理を繰り返す。

1.1 {初期設定} ~~min=?~~, no=? とおく。

1.2 {比較} $j=?$ から $??$ まで、次の処理を繰り返す

1.2.1 もし $a[j] < a[no]$ であれば、~~min=?~~, no=? とおく。

1.3 {交換} もし $no \neq i$ であれば $a[i]$ と $a[no]$ を交換

Question

- このアルゴリズムのどこをどう直せば降順／昇順ソートになる？

計算量

- ソートは比較と交換からなる
- 比較の回数:

$$(n - 1) + (n - 2) + \dots + 1 = \frac{(n-1+1)}{2} \times (n - 1) = O(n^2)$$

- 交換の回数:

最大でも(あるいは判定を使わなければ) $n-1$ 回で、 $O(n)$



- $O(n^2)$

第2回演習課題

1. 選択ソートアルゴリズムを実現する関数 `void SSort(int n, int a[])` を作成し、その動作を確認できるプログラム (`ex02-ssort-1.c`) を作成しなさい。ただし、`n` はデータの個数、`a[]` はデータ配列である。

実行例

```
./a.out
```

```
Input the number of data
```

```
5
```

```
Input the data
```

```
20 8 4 10 100
```

```
Sorted data
```

```
4 8 10 20 100
```

第2回演習課題

2. **(発展課題)** 最小値ではなく最大値を利用した昇順の選択ソートアルゴリズムを実現する関数 `void SSort(int n, int a[])` を作成し、その動作を確認できるプログラム (`ex02-ssort-2.c`) を作成しなさい。ただし、`n` はデータの個数、`a[]` はデータ配列である。