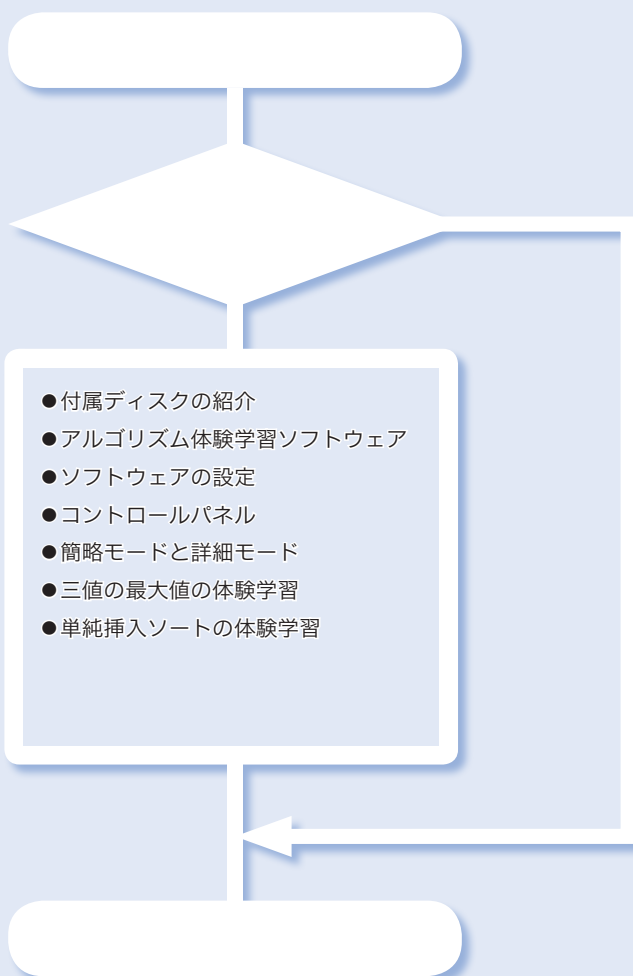


序章

アルゴリズム体験学習ソフトウェア



1

付属ディスクの紹介

■ 付属ディスクについて

さあ、アルゴリズムとデータ構造の学習を始めましょう。本書の学習で利用するプログラミング言語は、いまや世界中の幅広い分野で使われているC言語です。

さて、みなさんが、アルゴリズムやデータ構造を学習する目的は何でしょうか。思いつくところをいくつか挙げてみますと、

- C言語の基礎を習得した後に、次のステップを目指すため。
- 大学や専門学校の講義科目として。
- 資格取得のため。
- ...

といったところでしょうか。

目的が何であれ、本書の隅々まで理解することを目指しましょう。

もっとも、文字で書かれたテキストでの学習では、どうしても次のような問題にぶつかりがちです。

- 印刷されたプログラムリストや図は、静的であって“動き”がないため、それらを見るだけでは、アルゴリズムやデータ構造は、なかなか理解できるものではない。
- 利用するプログラミング言語に、それなりに精通する必要がある。
- アルゴリズムやデータ構造に関する基礎的な学習と、資格取得に要求される知識やテクニックは必ずしも一致しない。

このような問題を少しでも解決できるように、本書の付属ディスクには、数多くのソフトウェアやドキュメント類を収録しています。右ページに示すのが、収録内容の概略です。本書の学習とあわせて活用しましょう。

*

なお、序章である本章では、この中からアルゴリズム体験学習ソフトウェアについて簡単に紹介します。これ以外のソフトウェアやドキュメントについては、付録2 (p.417～) で紹介します。

■ 注意

付属ディスクに収録されているソフトウェアやドキュメント類の著作権は、著者が保有しています。これらを無断でコピーすることは禁じられており、法律に基づいて処罰の対象となることに注意してください。

また、ソフトウェアやドキュメント類の運用によって生じた損害などに対しては、一切の保証は行いませんので、あらかじめご了承ください。

■ 付属ディスクの収録内容の概略

ここでは、付属ディスクに収録しているソフトウェアおよびドキュメントの概要を示します。

■ アルゴリズム体験学習ソフトウェア

プログラムの実行に伴って、刻々と変化するプログラムの流れや変数の値などを、C言語で書かれたプログラムリストと対比しながら、視覚的に体験学習します。

- ▶ 既に数万人の方々に愛用されている学習ソフトウェアです。本書の学習と並行してご利用ください。

■ ソースプログラム

本文中に示している全 106 編のソースプログラムです。

- ▶ 学習内容を身に付けるためには、みなさん自身でプログラムを打ち込むのが基本です（特に初心者）。時間がない場合や、プログラムの動作を確認したい場合などに利用するとよいでしょう。

■ 演習問題の解答

本書の本文では、105 問の演習問題を出題しています。すべての問題の解答を収録しています。

- ▶ 演習問題は、みなさんが自分自身で解く、というのが原則です。まずは自分で解いてみて、分からない場合などに参照するようにしましょう。

■ 入門書コーナー

C 言語と基本情報技術者試験に関するテキストの 325 ページ分を収録しています。文書の形式は PDF であるため、本のレイアウトなども、そっくり再現されます。Adobe Reader などのソフトウェアで閲覧できます。

- ▶ 収録しているのは、各テキストの一部です。書籍の全体を読みたい場合は、書店などで購入いただくと幸いです。

■ 基本情報技術者試験の過去問題と解説

平成 6 年度秋期から平成 22 年度秋期までの基本情報技術者試験（旧・第 2 種情報処理技術者試験）の問題から、アルゴリズムおよびデータ構造に関連する分野の午前試験の問題（全 165 問）と解答・解説を収録しています。

PDF 形式と HTML 形式の二つの形式の好きなほうでご覧ください。Internet Explorer や Firefox、Opera、Safari、Google Chrome などの WWW ブラウザで閲覧できます。

- ▶ 本書各章の章末問題で出題しているのは、基本情報技術者試験の問題です。

2

アルゴリズム体験学習ソフトウェア

■ アルゴリズム体験学習ソフトウェアについて

ソフトウェアを実行すると、その内部では、プログラムの流れの分岐や繰返しなどが行われますし、変数の値は刻々と変化していきます。

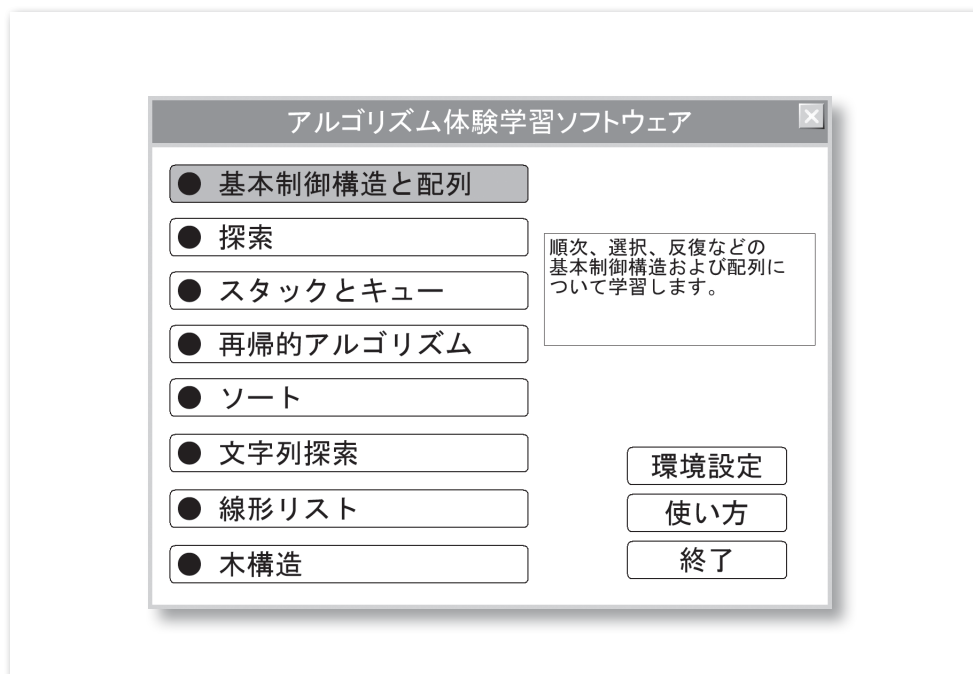
そのため、紙に印刷された、動きのない静的なプログラムリストや図では、ダイナミックに変化するプログラムの動きを理解するのは容易ではありません。

付属ディスクに収録しているアルゴリズム体験学習ソフトウェアは、本書で紹介するアルゴリズムやデータ構造のもつ動きを、C言語のプログラムリストや解説などと対比しながら視覚的に体験学習できるようにしたものです。

- ▶ 本ソフトウェアは、Microsoft Windows XP / Vista / 7上で動作します。本ソフトウェアを実行するには、事前にインストールを行う必要があります。インストールの方法は付録2 (p.420)で解説しています。

■ メニュー画面

アルゴリズム体験学習ソフトウェアを起動しましょう。そうすると、**Fig.1**に示すメニュー画面が表示されます。

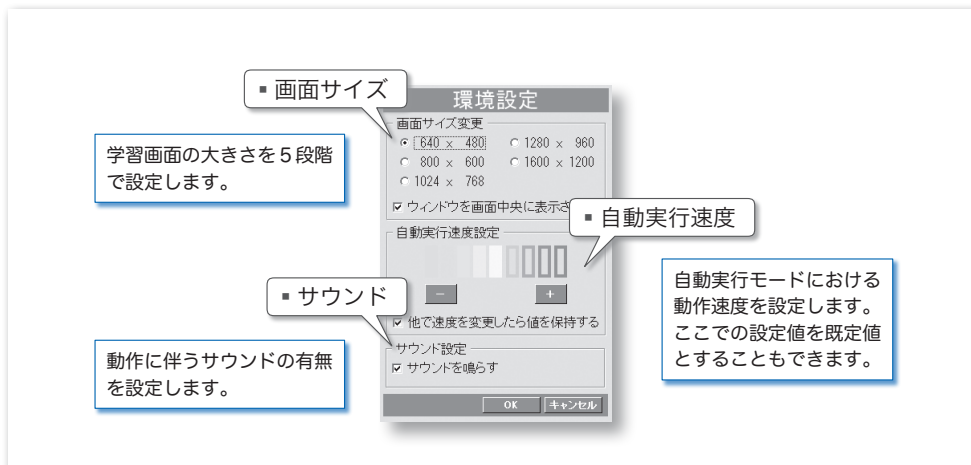


● Fig.1 アルゴリズム体験学習ソフトウェアのメニュー画面

■ 動作環境の設定

体験学習を始める前に、まずは環境設定を行いましょう。

メニュー画面から〈環境設定〉を選んでください（マウスカーソルを位置付けてから左ボタンをクリックします）。そうすると、**Fig.2** に示すウィンドウが表示されます。



● **Fig.2** 環境設定の画面

■ 画面サイズ

本ソフトウェアは、実行する環境のディスプレイの解像度や大きさに依存することなく快適に利用できるように作られています。みなさんの環境・好み・視力などにあわせて、画面サイズを選択しましょう。

また、本ソフトウェアをディスプレイ画面の中央に表示するかどうかも選択できます。

■ 自動実行速度

本ソフトウェアの実行モードには、〈ステップ実行〉と〈自動実行〉との2種類があります（これらのモードについては、次ページで解説します）。ここで設定するのは、自動実行モードでの動作速度の既定値です。体験学習をしていて、自動実行のスピードが速い、あるいは遅いと感じたら、この値を調整するとよいでしょう。

また、体験学習中に自動実行速度の調整を行った場合に、それを既定値として記憶させるかどうかも選択できます。

■ サウンド

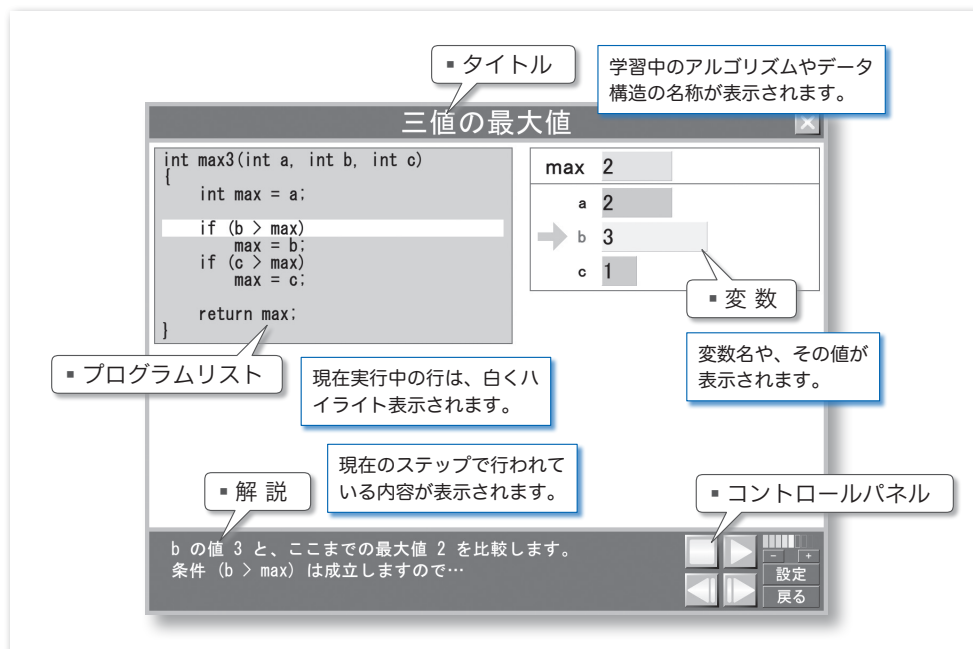
動作に伴う効果音のON/OFFを設定します。

*

設定が終了したら、いくつかのアルゴリズムを体験学習してみましょう。

■ <三値の最大値>の体験学習

メニュー画面で<基本制御構造と配列>を選び、さらに<三値の最大値>を選んでみましょう。そうすると、**Fig.3** に示す画面が表示されます。



● **Fig.3** 三値の最大値の学習画面

ここでは、第1章で学習する『三値の最大値を求める』アルゴリズムを体験学習します。ここに示されている関数 `max3` は、受け取った三つの仮引数 `a`, `b`, `c` の最大値を求めて変数 `max` に格納して、その値を返却する関数です。

ウィンドウには、プログラムリスト、解説、変数などが表示されています。慣れるまでは少々大変でしょうが、各項目を見比べながら学習を進めていきます。

本ソフトウェアの操作に利用するのが、ウィンドウの右下隅に表示されているコントロールパネルです。**Fig.4** を見ながら理解していきましょう。

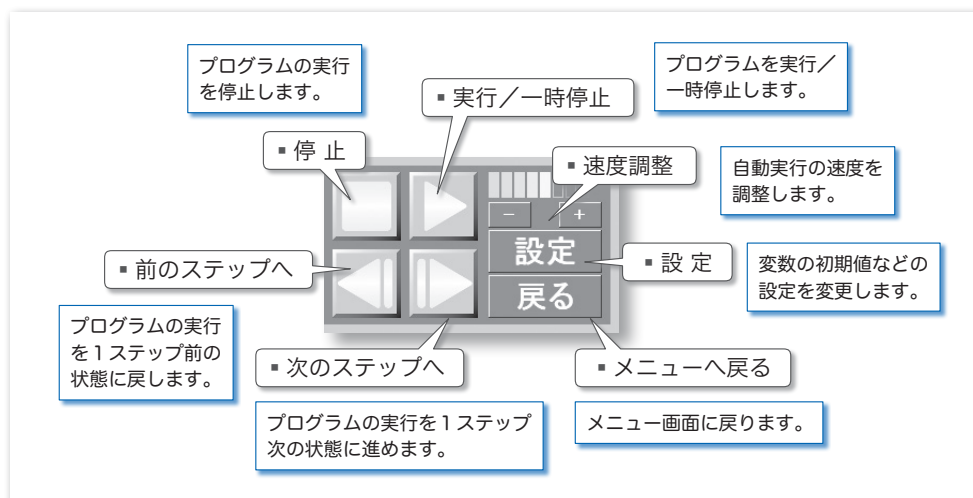
■ 停止

プログラムの実行を停止します。

■ 実行/一時停止

プログラムを自動実行モードで実行します。このモードでは、停止や一時停止をしない限り、プログラムが1ステップずつ最後まで実行されます。

なお、実行中は、動作を一時停止するためのボタンとして機能します。



● Fig.4 コントロールパネル

■ 速度調整

自動実行モードにおいて、プログラムを1ステップ進めるスピードの調整を行います。+をクリックすると速くなり、-をクリックすると遅くなります。

■ 次のステップへ

プログラムをステップ実行モードで1ステップずつ実行します。プログラムリストのどこが実行されているか、変数の値がどのように変化するか、何が行われているのかを確認しながら理解を深めます。

■ 前のステップへ

プログラムの実行を1ステップ前に戻します。プログラムの動きや変数の値の変化を見過ごしたときなどに使います。戻れるステップ数に制限はありません。プログラムの開始状態まで戻ることができます。

■ 設定

同じプログラムでも、変数の初期値などの条件が異なれば、その流れや、得られる結果、終了までに要するステップ数や時間などが変化します。

Fig.5 に示すウィンドウが表示されますので、変数 a, b, c の初期値を変更してみましょう。

どの組合せでも、正しく最大値を求めることができるのでしょうか？ 三値のどれが最大値になるのでしょうか？ いろいろと試してみましょう。



● Fig.5 設定画面

■ <単純挿入ソート>の体験学習

次に、第6章の<単純挿入ソート>を体験学習しましょう。いったんメニューに戻って<ソート>を選び、それから<単純挿入ソート>を選んでください。

- ▶ ソートとは、データの集まりを小さい順や大きい順に並べることです。単純挿入ソートのアルゴリズムは、6-4節で学習します。

■ 簡略モード

この体験学習画面では、要素数が10である配列aを昇順にソートしていく過程が示されます。右側の白いウィンドウには、配列の要素の値が、ちょうど棒グラフのように表示されます。なお、棒の左側に小さく表示された0から9の値は各要素の添字の値であり、棒の中に表示された値が各要素の値です。

単純挿入ソートは、変数iの値を1から始めて一つずつ増やしながら、着目している要素a[i]を、それより先頭側の“適当な位置に挿入する”ことによって、配列の要素を並べかえるアルゴリズムです。

プログラムを実行すると、**Fig.6**に示すように、要素が挿入される様子がアニメーション表示されます。

- ▶ ここに示しているのは、1と5のあいだに3が挿入される様子です。

```
void insertion(int a[], int n)
{
    int i;
    for (i = 1; i < n; i++) {
        int tmp = a[i];
        tmpをa[0]...a[i-1]の適当な位置に挿入
    }
}
```

0 1

1 3

2 5

3 7

4 9

5 8

6 6

7 10

8 4

9 2

比較回数	引数	変数
3	n: 10	i: 3
交換回数		
3		

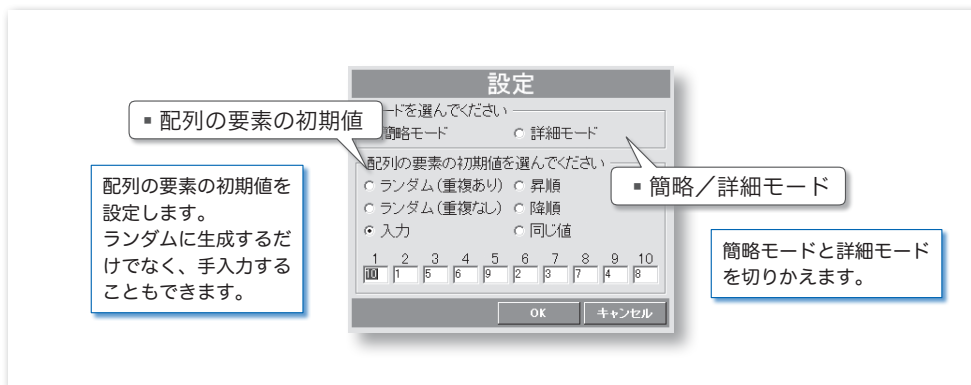
tmpの値 3 を a[3] より先頭側の適当な位置に挿入します。
ここでは、a[1] の位置に挿入します。

● Fig.6 単純挿入ソートの簡略モード

まず $a[1]$ が挿入され、次に $a[2]$ が挿入されて、…、最後に $a[9]$ が挿入されると、配列の要素は、昇順（小さいほうから順）に並びます。これで、ソートは完了です。

*

このアルゴリズムが大まかに理解できたら、<設定>ボタンを押してください。**Fig.7** の画面が表示されます。



● **Fig.7** 単純挿入ソートの設定画面

ソートの対象となる配列要素の初期値が自由に設定できます。初期値を変更すると、ソートが終了するまでに行われる**比較回数**や**交換回数**は、どのように変化するでしょう。

たとえば、<昇順>を選んでみてください。そうすると、もともとソート済みの配列をソートすることになります。このような場合は、非常に少ないステップ数で、ソートが終了します。逆に<降順>だと、どうなるでしょうか。

また、個々の要素の初期値をキーボードから入力することもできます。いろいろと試してみましよう。

■ 簡略モードと詳細モード

ここまでは、アルゴリズムの概略を学習するための**簡略モード**でした。このモードでは、ソートの過程における“**適当な位置に挿入する**”処理が、プログラムリスト上、単一のステップとして行われています。

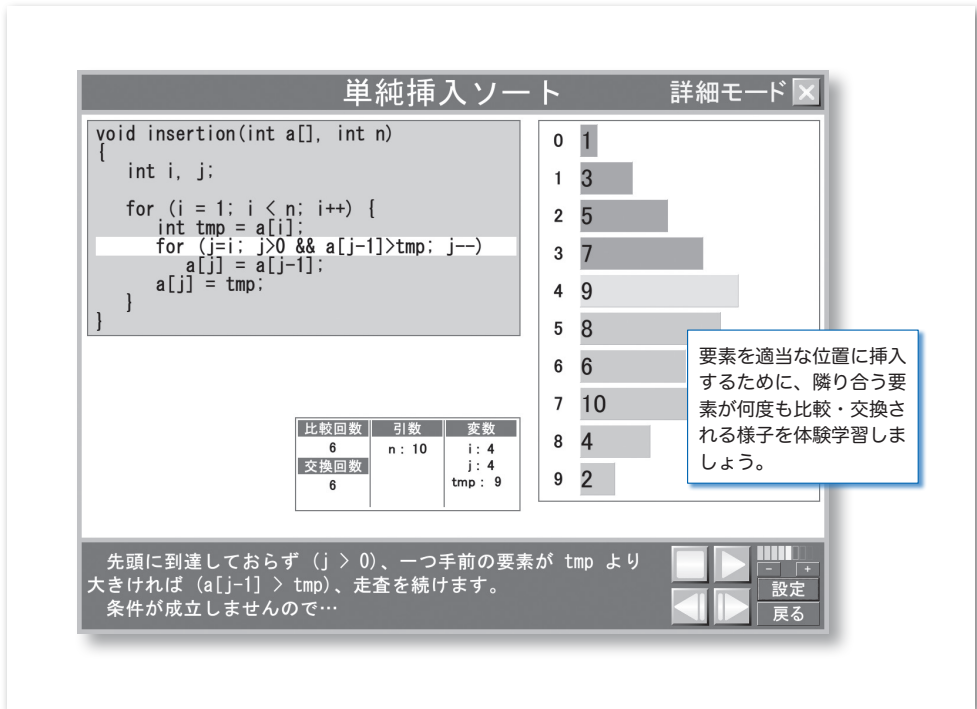
しかし、配列内の要素を“**適当な位置に挿入する**”という処理は、実際のプログラムでは単一の文や命令で実現できるものではありません。

第6章で詳しく学習しますが、配列を先頭側へと走査しながら（なぞっていきながら）、一つ前の要素が $a[i]$ より大きければ交換する、という地道な作業を、 $a[i]$ 以下の値に出会うまで繰り返す必要があります。

その詳細までも完全に実現したプログラムを学習するのが、**詳細モード**です。設定画面から、<詳細モード>を選んでみましょう。

■ 詳細モード

詳細モードでは、**Fig.8** に示すように、完全な形で実装されたプログラムリストと対比しながら体験学習を行います。



● **Fig.8** 単純挿入ソートの詳細モード

配列要素 $a[i]$ の“適当な位置への挿入”が、代入作業の繰返しによって行われることを理解しましょう。

挿入ソートを含めて、いくつかのアルゴリズムでは、簡略モードと詳細モードの両方が用意されています。

簡略モードは、アルゴリズムの根本的な考え方を学習するモードです。まずは、こちらのモードでの学習を行うとよいでしょう。

詳細モードは、完全な形のプログラムリストを学習するモードです。完全なプログラムの動作が学習できるというメリットがある一方で、ステップ数が非常に多くなります（その分、クリック数も増えます）。

みなさんの理解度などに応じて、モードを切りかえて学習を進めてください。

*

それでは、<戻る>ボタンを押して、ソートのメニューに戻ってみましょう。ソートを行うために、数多くのアルゴリズムが考案されています。これらのアルゴリズムだけでなく、いろいろなソートアルゴリズムの速度比較なども体験学習してみましょう。

その他のアルゴリズムの体験学習

ここでは、三値の最大値を求めるアルゴリズムと、単純挿入ソートを例にして、アルゴリズム体験学習ソフトウェアの利用法を簡単に紹介しました。これ以外にも、多くのアルゴリズムやデータ構造を体験学習できますので、いろいろと試してみてください。

ハノイの塔

```
void move(int no, int x, int y)
{
    if (no > 1)
        move(no - 1, x, 6 - x - y);
    printf("[%d]を%d軸から%d軸へ%N", no, x, y);
    if (no > 1)
        move(no - 1, 6 - x - y, y);
}
```

no : 2

重ねられた円盤を最短の回数で別の軸へ移動するアルゴリズムです。

円盤 2 を、第 1 軸か

かまかせ法

```
int bf_match(char txt[], char pat[])
{
    int pt = 0;
    int pp = 0;
    while(txt[pt]!='\0' && pat[pp]!='\0') {
        if (txt[pt] == pat[pp]) {
            pt++;
            pp++;
        } else {
            pt = pt - pp + 1;
            pp = 0;
        }
    }
    if (pat[pp] == '\0')
        return pt - pp;
    return -1;
}
```

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
A B C A C B A A B A C A B C
| x
A B A C
0 1 2 3

変数
pt : 4 pp : 1
比較回数
7

かまかせ法は、文字列の探索を行うアルゴリズムです。

ソート速度比較 2

```
void bubble(int a[], int n)
{
    int i, j;
    for (i = 0; i < n - 1; i++) {
        for (j = n - 1; j > i; j--)
            if (a[j - 1] > a[j])
                swap(int, a[j - 1], a[j]);
    }
}
```

```
void bubble(int a[], int n)
{
    int i, j;
    for (i = 0; i < n - 1; i++) {
        int exchg = 0;
        for (j = n - 1; j > i; j--)
            if (a[j - 1] > a[j]) {
                swap(int, a[j - 1], a[j]);
                exchg++;
            }
        if (exchg == 0) break;
    }
}
```

```
void bubble(int a[], int n)
{
    int k = 0;
    while (k < n - 1) {
        int j;
        int last = n - 1;
        for (j = n - 1; j > k; j--)
            if (a[j - 1] > a[j]) {
                swap(int, a[j - 1], a[j]);
                last = j;
            }
        k = last;
    }
}
```

0:00.00 0:00.00 0:14.96

ここに示しているのは一例です。たくさんのアルゴリズムやデータ構造をみなさん自身で操作しながら体験学習しましょう。

単純交換ソートアルゴリズムと二つの改良版の速度を比較します。

実行中です。

設定 戻る

● Fig.9 アルゴリズム体験学習画面の一例