

アルゴリズムとデータ構造を  
**Excel VBA**で学ぶ

猪股俊光 [共著]  
呉 書雅

コロナ社

# まえがき

デジタル時代に必要な素養として情報教育が重要視され、そのための基礎的一分野として、数理・データサイエンス・AIに関連した教育が活発化されている。なかでも「プログラミング」は小中高でプログラミング教育が導入されていることもあり、文系・理系を問わず大学の情報関連科目でも取りあげられている。プログラミング学習用の書籍も多く存在しているが、CやC++など、一般的な企業の日常業務ではあまり利用されていないシステム開発用のプログラミング言語によるものがほとんどである。

そこで、本書では、日常業務での活用につなげられるよう、企業や学校、官公庁などの業務で広く利用されている表計算ソフトの一種である Excel（およびそのプログラミング言語 VBA）を用いてプログラミングの基礎を学べる内容とした。プログラミングの学習は難しいと思っている読者も多いだろう。その要因の一つとして、プログラムが実行されている様子が目に見えないことがあげられる。もし、プログラムを実行したときにデータがどのように処理されていくのか、多少なりとも目視できれば、学習の助けとなろう。さらに、プログラムがデータを処理していく様子を頭のなかでイメージ化できるようになれば、正しいプログラム（望みの結果を出力できるプログラム）を作成する力の向上につなげられるだろう。

そこで、本書では、表計算ソフトのワークシートを記憶装置（メモリ）と見立て、プログラムの実行中にメモリが更新される様子を、ワークシートの内容が更新されることに対応させた説明やプログラム例を導入した。加えて、本書はつぎの特徴をもつ。

## 本書の特徴

- プログラミングを学ぶうえで、必要とされるハードウェアやソフトウェアの基礎知識から、プログラミング能力向上のための基盤となる「アルゴリズムとデータ構造」を体系的に学べる内容とした。これらは、文部科学省「数理・データサイエンス・AI教育プログラム認定制度（MDASH）」のカリキュラムの一部（応用基礎レベルとリテラシーレベルのプログラミング基礎関連項目など）に該当する。
- データサイエンスやAIの日常業務への展開を見据えて、データサイエンスとAIについて表計算ソフトを使ったプログラム例も交えながら説明した。
- プログラミングの初学者を想定しつつ、限られた機能だけを使ったプログラム作りから始め、徐々に多くの機能を使ったプログラム作りへと発展していく内容とした。その過程において、説明内容の理解を深めるための例を提示するとともに、理解度を確かめるための章末問題を掲載した。なお、その解答例は本書のサポートページに掲載している。

- 表計算ソフトは、Microsoft 365 (Windows 版) とし、開発環境 VBE のもとで、プログラミング言語 VBA を使ったプログラムの編集・実行・デバッグの方法について述べた。特に、デバッグのための機能については、プログラムが実行されている様子を目で確かめるために有効であることから、本書の前半 (第 3 章) で取りあげた。

なお、macOS 版 Microsoft 365 を利用する読者のために、本書のサポートページにおける付録にて、Windows 版との違いについて説明した。

本書の第 1 章～第 12 章 (担当 猪股) は、岩手県立大学全学共通科目『データサイエンス応用 I』と岩手県立大学盛岡短期大学部生活科学科『情報処理演習』のために作成した講義資料を原案とし、これまでの講義実践をもとに書き直すとともに、新たに第 13 章と第 14 章 (担当 呉) を加筆した。なお、『データサイエンス応用 I』は岩手県立大学の「文理融合データサイエンス教育プログラム」の応用基礎レベル (令和 6 年度 MDASH 認定) 対応科目の一つであり、『情報処理演習』は盛岡短期大学部の「文理融合データサイエンス教育プログラム」のリテラシーレベル (令和 5 年度 MDASH 認定) 対応科目の一つである。

末筆ながら本書の作成にあたっては、『データサイエンス応用 I』や『情報処理演習』の受講生をはじめ、ご協力をいただいた方々に感謝する。特に、岩手県立大学ソフトウェア情報学部 4 年生 (2025 年 8 月現在) の平彩斗氏、鷹觜陸氏、橋端瞭氏には、本書の内容を学生の目線で確認していただいた。さらに、出版にあたっては株式会社コロナ社の皆さまにご尽力いただいた。皆さま方にこの場をかりてお礼申し上げる。

2025 年 8 月

猪股俊光  
呉 書雅

### 本書のサポートページ (コロナ社ホームページ)

本書における付録や章末問題の解答例は以下のページに掲載している。

[https://www.coronasha.co.jp/np/resrcs/docs.html?goods\\_id=8816](https://www.coronasha.co.jp/np/resrcs/docs.html?goods_id=8816)

解凍用パスワード :



# 学習の手引き

## 本書の構成と読者

本書では、全体を 14 章に分け、それらを図 1 に示す関連性をもたせて編集した。

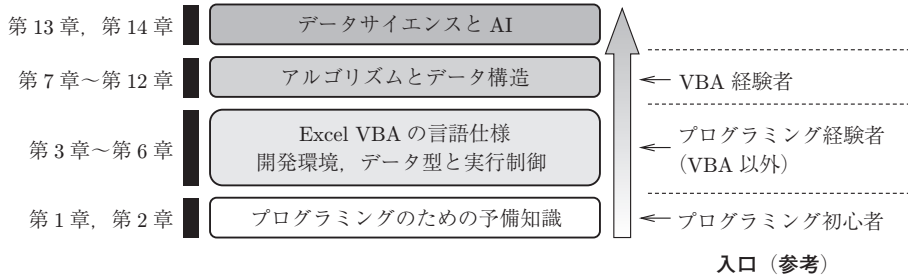


図 1 各章の構成

第 1 章と第 2 章ではプログラミングを学ぶ際に予備知識として求められる内容について述べ、つづく第 3 章～第 6 章では Excel VBA をもとに、プログラミング言語の用語・構文・演算子・制御構造、開発環境を述べた。

そして、第 7 章～第 12 章では「アルゴリズムとデータ構造」の基礎概念・プログラミング技法・アルゴリズムの評価法について述べた。Excel VBA を用いて説明しているが、他言語でプログラムを記述する場合にもあてはまる基礎内容である。

さらに、第 13 章と第 14 章は、これからの社会の発展に期待されている「データサイエンスと AI」についてプログラミングの観点から述べた。

このように、本書は、プログラミング初心者を念頭に、第 1 章から順に読み進められることを想定した構成としている。そのため、図 1 において、「プログラミング初心者」の入口として第 1 章と第 2 章が指し示されている。

もし、Excel VBA 以外の言語でプログラミングの経験があり、コンピュータのハードウェアとソフトウェアの知識がある読者であれば、第 3 章から読み進めてもよいだろう。その場合、使用経験のある言語と Excel VBA の違いに注目して読んでほしい。

そして、Excel VBA によるプログラミング経験がすでにある場合には、第 7 章から読み進めることが想定される。

## プログラムの表記

本書では Excel VBA のプログラムを図 2 のように記す。

太字の単語はキーワードと呼ばれる Excel VBA の構文要素であり、これらの単語は変数などの名前として用いることはできない。行番号はプログラムの説明の都合上追記したもので、

```

行番号
1. Sub average() コメント
2.   Dim x As Integer, y As Integer ' Step.1 変数の使用宣言
3.   x = InputBox("x=") ' Step.1 データ x の入力
4.   y = InputBox("y=") ' Step.1 データ y の入力
5.   MsgBox "平均値=" & (x + y) / 2 ' Step.3 平均値の出力
6. End Sub
空白文字  
(スペース)
キーワード

```

図 2 プログラムの表記例

プログラムの一部ではない。「'」で始まる行末までの文字列はコメント（注釈）であって、いずれもプログラムの一部ではなく、本書ではグレーで表記している。「」は空白文字（スペース）を表す。文字列のなかや、入力時に見落としを防ぐために用いる。なお、本文中では単に空白と記す。

### 動作環境

本書の実行例の動作環境は表のとおりである。

表 本書の動作環境

OS	Windows 11 Pro
Excel	Microsoft 365 Apps for enterprise

本書のプログラムを、macOS 版 Microsoft 365 において実行する場合の留意事項については、本書のサポートページに掲載している（詳しくはまえがき末尾をご覧ください）。

### さらなる学びのために

本書は、プログラムの記述言語と実行環境として Excel VBA と VBE を使用しているが、それらの解説書ではない。「アルゴリズムとデータ構造」の説明に必要な範囲で Excel VBA のオブジェクトや関数などを取りあげた。そのため、Excel VBA が有するセルやワークシートの高度な操作、図形やグラフの操作、ユーザーフォームの作成などについては Excel VBA の解説書を参照されたい。

本書の読者のなかには、将来的に Excel VBA 以外の言語でプログラムを作成する機会がある人がいるかもしれない。そのときには、本書の第 3 章～第 6 章で述べている「定数・変数・データ型の種類、制御文の種類、関数の定義の仕方」などが、新たに使用するプログラミング言語においてどう定められているのかに着目しながら、その言語を習得してほしい。

# 目 次

## 1. コンピュータの世界

1.1	コンピュータはことばで動く	1
1.2	コンピュータの動き	2
1.2.1	コンピュータの中身	2
1.2.2	プログラムの実行とメモリ	3
1.2.3	CPUの命令の種類	3
1.2.4	命令実行サイクル	3
1.3	デジタルの世界	4
1.3.1	2進法によるデジタル化	4
1.3.2	数のデジタル化	5
1.3.3	文字のデジタル化	6
1.3.4	画像のデジタル化	6
1.3.5	音のデジタル化	8
	章末問題	9

## 2. プログラムの世界

2.1	ソフトウェアの役割	10
2.2	アルゴリズムとプログラム	11
2.3	プログラムの変数とメモリ	12
2.4	プログラムの開発過程	13
2.5	オブジェクト指向プログラミング	14
2.5.1	VBAとオブジェクト指向	14
2.5.2	プログラムのなかのオブジェクト	14
	章末問題	15

## 3. VBAと開発環境

3.1	表計算ソフトとマクロ	16
3.2	VBAの開発環境VBE	16
3.2.1	Excelの構成要素	16
3.2.2	プログラムの開発環境VBE	17
3.2.3	プログラムの開発過程	18

3.3	プログラムの編集	19
3.4	ファイルの保存・読み込み	20
3.4.1	ファイルの保存	20
3.4.2	ファイルの読み込み	20
3.5	プログラムの実行	20
3.5.1	VBEのもとでの実行	20
3.5.2	ワークシートのもとでの実行	21
3.6	VBEによるデバッグ	22
3.6.1	デバッグの概要	22
3.6.2	実行時エラーのときのデバッグ	23
3.6.3	計算間違いのときのデバッグ	25
3.7	デバッグをするときのポイント	28
	章末問題	29

## 4. ワークシートの基本操作

4.1	Excelのオブジェクト	30
4.2	オブジェクトのプロパティの利用	31
4.2.1	プロパティの形式	31
4.2.2	セルへの入力	32
4.2.3	オブジェクトの省略	34
4.2.4	複数のワークシートのセルの 計算	34
4.3	オブジェクトのメソッドの利用	35
4.3.1	メソッド	35
4.3.2	マクロの記録	37
4.4	VBAの組み込み関数	38
4.4.1	関数	38
4.4.2	数値関数	38
4.4.3	文字列関数	39
4.4.4	Excelの関数の利用	40
4.5	イミディエイトウィンドウの使い方	40
4.5.1	プログラムの実行中の利用	41
4.5.2	対話形式での利用	42
	章末問題	43

<b>5. プログラムの構造とデータ型</b>	
5.1 VBA のプログラムの構造	44
5.2 データ型	45
5.2.1 データ型の値の範囲	45
5.2.2 配列	46
5.3 変数の宣言と利用	46
5.3.1 宣言された変数	46
5.3.2 宣言なしの変数：Variant 型	47
5.3.3 変数の明示的宣言：Option Explicit	48
5.4 データ型と計算	49
5.4.1 Integer 型や Double 型の計算	49
5.4.2 Boolean 型の計算	50
5.4.3 String 型の計算	51
5.4.4 データ型と演算や代入	53
5.5 関数の定義と呼び出し	53
5.5.1 ユーザ定義の関数	53
5.5.2 関数呼び出しの実引数	55
章末問題	56
<b>6. プログラムの実行制御</b>	
6.1 実行制御の種類	58
6.1.1 逐次処理	58
6.1.2 実行制御の種類	58
6.2 流れ図	59
6.2.1 流れ図のブロック	59
6.2.2 流れ図と実行制御	59
6.3 条件分岐	61
6.3.1 If 文	61
6.3.2 Select 文	62
6.4 繰り返し処理	63
6.4.1 For 文	63
6.4.2 While 文	65
6.4.3 繰り返しの強制終了	67
6.5 関数呼び出しによる繰り返し処理	67
6.6 プログラムのなかのプロシージャと変数	69

6.6.1 プロシージャの集まりとしての プログラム	69
6.6.2 引数をもつプロシージャの定義と 利用	70
6.6.3 大域変数と局所変数	70
章末問題	72

## 7. アルゴリズムと計算量

7.1 アルゴリズムの評価	73
7.2 アルゴリズムの複雑さ	76
7.2.1 計算量	76
7.2.2 最悪計算量と平均計算量	76
7.2.3 オーダ記法 $O$	77
7.3 計算量の改善策	78
7.3.1 総和計算の時間計算量	78
7.3.2 べき乗計算の時間計算量	79
7.3.3 再帰関数の領域計算量	80
7.3.4 末尾再帰呼び出しによる領域計算量 の改善	80
章末問題	82

## 8. 探索のアルゴリズム

8.1 探索の概要	83
8.1.1 探索の成功と失敗	83
8.1.2 VBA での実装法	83
8.2 線形探索	84
8.3 2分探索	85
8.4 探索プログラムの実行	88
8.5 ハッシュ法による探索	90
8.5.1 ハッシュ法の概要	90
8.5.2 開アドレス法	90
8.5.3 連鎖法	91
章末問題	92

## 9. 整列のアルゴリズム

9.1 整列の概要	93
9.1.1 順序関係による並び替え	93
9.1.2 実装上のテクニック	93

9.2	バブルソート	95
9.2.1	基本的なアイデア	95
9.2.2	バブルソートのプログラム	95
9.3	クイックソート	98
9.3.1	基本的なアイデア	98
9.3.2	クイックソートのプログラム	98
9.4	特色のある整列法	101
9.4.1	バケットソート： $O(n)$	101
9.4.2	マージソート：大規模データ	102
	章末問題	102

## 10. リスト構造とアルゴリズム

10.1	データ構造の必要性	104
10.1.1	これまでのデータ処理の課題	104
10.1.2	ワークシートを使ったデータ構造 の実現	104
10.2	リスト構造	105
10.2.1	リストと操作	105
10.2.2	リストのためのデータ構造	105
10.2.3	ワークシート上での実装法	106
10.3	スタック	108
10.3.1	データの追加：push	108
10.3.2	データの削除：pop	109
10.4	キュー	110
10.4.1	データの追加：enqueue	110
10.4.2	データの削除：dequeue	111
10.5	リスト構造のもとでの探索と整列	112
10.5.1	リスト構造のもとでの探索	112
10.5.2	リスト構造のもとでの整列	112
	章末問題	114

## 11. 木構造とアルゴリズム

11.1	2分木とデータ構造	115
11.1.1	2分木の概要	115
11.1.2	2分木の実装法	116
11.2	2分木の操作と整列	117
11.2.1	データ追加の方法	117
11.2.2	データ追加のプログラム	118
11.2.3	データ削除の方法	119
11.2.4	2分木の整列と探索	120

11.3	木の探索	121
11.3.1	探索の方法	121
11.3.2	迷路の探索の実装法	122
11.3.3	深さ優先探索	123
11.3.4	幅優先探索	125
	章末問題	126

## 12. ファイルとデータベース

12.1	ファイル操作の準備	127
12.1.1	ファイルの構造	127
12.1.2	CSVファイル	127
12.2	ファイルの操作	128
12.2.1	ファイルのパス	128
12.2.2	ファイルの基本操作	128
12.2.3	CSVファイルの読み込み	129
12.2.4	CSVファイルへの書き込み	131
12.3	ワークシートの操作	132
12.3.1	レコードの整列	132
12.3.2	ワークシートへの分配	134
12.3.3	データの分析	136
12.4	関係データベース	137
12.4.1	関係データベースの演算	137
12.4.2	レコードの射影	137
12.4.3	レコードの選択	138
12.4.4	ワークシートの結合	140
	章末問題	141

## 13. データサイエンスの世界

13.1	データサイエンス	142
13.2	ビッグデータ	143
13.2.1	ビッグデータの定義と特徴	143
13.2.2	データの構造	143
13.3	ビッグデータの収集と蓄積	144
13.3.1	データ収集の方法	144
13.3.2	データ蓄積の方法	145
13.4	クラウドサービスとその役割	145
13.4.1	クラウドの基本概念	145
13.4.2	ビッグデータ処理における利点	145
13.5	ビッグデータの活用事例	146
13.6	オープンデータ	147

13.7 表計算ソフトとデータサイエンス …… 148  
章 末 問 題 …… 149

## 14. AI の 世 界

14.1 AIとプログラミング …… 150  
14.1.1 AIの歴史の概略 …… 150  
14.1.2 旅行先案内システム …… 150  
14.1.3 コンピュータとの対話システム …… 152  
14.2 AI活用の留意事項 …… 154

14.3 生 成 AI …… 155  
14.3.1 生成 AIの基盤モデル …… 155  
14.3.2 生成 AIの応用分野 …… 156  
14.3.3 プロンプトエンジニアリング …… 157  
章 末 問 題 …… 158

引用・参考文献 …… 159  
あ と が き …… 162  
索 引 …… 163

## 例 8.2 2分探索

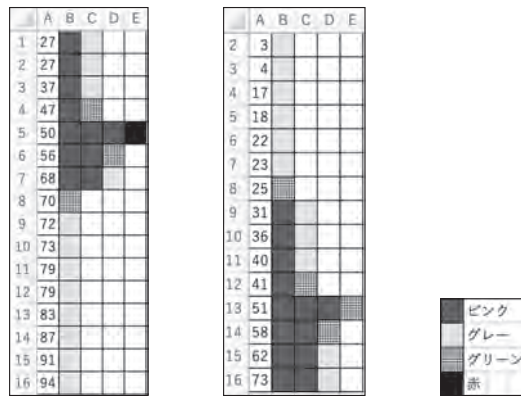
2分探索では、探索対象のデータをあらかじめ整列しておく必要がある。そのための方法の一つとして、Range オブジェクトのメソッド Sort の利用法を示す。そのために、メインプロシージャにあたる search\_main\_binary() では、4 行目で A 列のセル (A16 まで) を昇順に整列するために、Range オブジェクトのメソッド Sort を利用している (2~3 行目は search\_main\_linear() と同じ)。そして、5 行目で binary\_iter() をキーを 50 として呼び出す。

```

1. Sub search_main_binary()
2.   Call Clear_Cells("A1:Z16")
3.   Range_Rnd ("A1:A16")
4.   Range("A1:A16").Sort Key1:=Range("A1") ' 2分探索の場合
5.   Debug.Print binary_iter(50, 1, 16) ' 2分探索の場合
6. End Sub

```

データは 1~100 の乱数によるものとしたため、実行のたびに結果が異なる。図 8.2 は 2 回の実行結果である。同図から探索の範囲が半分ずつ減っていくことが見てとれる。プロシージャ mark\_cells() では、ピンクのセルは探索範囲、グレーのセルは探索範囲外、グリーンは中央のセル、赤は見つかったキーのセルとしてそれぞれ描く。しかしながら、図 8.2 では凡例のようにセルの背景のパターンの違いでそれらを表している。



(a) 実行例 1 (b) 実行例 2

図 8.2 2分探索の実行例

実行例 1 では、中央のセル (B8) の値 70 が 50 よりも大きいため、つぎに前半の 7 個 (A1:A7) を探索し、そのつぎには後半の 3 個 (A5:A7) を探索し、と続けていき、4 回目に 50 (E5) を見つけ、成功している。

また、実行例 2 では、中央のセル (B8) の値 25 が 50 よりも小さいため、つぎに後半の

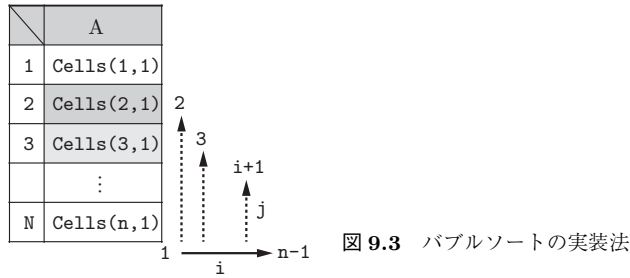


図 9.3 バブルソートの実装法

変数  $c$  は交換の回数 (`swap()` の呼び出し回数) を数え上げるためのものであり、その値は 25 行目でイミディエイト ウィンドウに出力される。

例 9.2 バブルソート

$N = 8$  として、A 列に 8 個の乱数 (1~100) を格納したうえで、`bubble_sort()` を呼び出すのがつぎのプログラムである (`Clear_Cells()` や `Range_Rnd()` は 8.4 節参照)。

```

1. Sub Test_sort()
2.   Call Clear_Cells("A1:Z8") ' セルをクリア (視覚化の場合)
3.   Range_Rnd ("A1:A8") ' 乱数を格納
4.   Call bubble_sort(8) ' アルゴリズム開始
5.   Range("A1:A8").Interior.Color = rgbLightBlue
6. End Sub
    
```

5 行目では整列されたあとの A 列の背景色を青色にしている (B 列は初期状態での並びにあたる)。この結果、整列される様子が図 9.4 のように視覚化される。ピンク色は交換対象となったセルであり、小さい数がだんだんと浮かび上がる様子が見てとれる。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	15	25	25	25	25	25	25	15	15	15	15	15	15	15	15	15
2	25	55	55	55	55	55	15	25	25	25	25	25	25	25	25	25
3	29	62	62	62	62	15	55	55	55	55	55	29	29	29	29	29
4	52	68	68	68	15	62	62	62	62	62	29	55	55	55	52	52
5	55	52	52	15	68	68	68	68	68	29	62	62	62	52	55	55
6	62	99	15	52	52	52	52	52	29	68	68	68	52	62	62	62
7	68	15	99	99	99	99	99	29	52	52	52	52	68	68	68	68
8	99	29	29	29	29	29	29	99	99	99	99	99	99	99	99	99

図 9.4 バブルソートの実行例

このアルゴリズムの計算量は、整列対象のデータの総数が  $N$  であるとき、プロシージャ `bubble_sort()` の 12 行目での比較回数をともに求めてみると、10 行目と 11 行目の 2 重の For 文から、式 (9.2) より  $O(N^2)$  であることがわかる。

$$(n - 1) + (n - 2) + \dots + 2 + 1 = \sum_{k=1}^{N-1} k = \frac{N(N - 1)}{2} \tag{9.2}$$

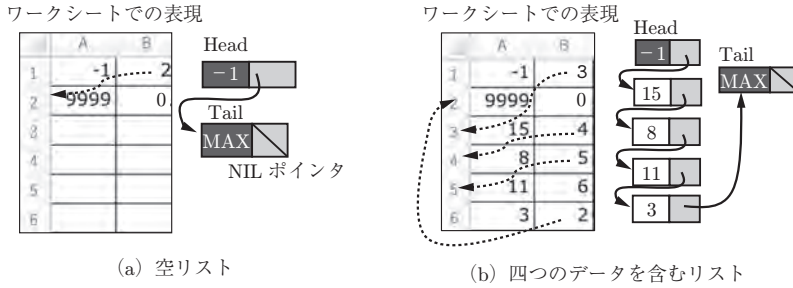


図 10.5 リスト構造のワークシートでの実現例

さらに、リストを実現するために VBA のつぎの機能を使うことにする。

- 定数を定義する プログラムのなかで特定の定数 *val* (データ型は *type*) に、つぎの構文 (Const ステートメント) で名前 *name* をつけることができる (データ型の指定は省略可)。

```
Const name As type = val
```

そこで、番兵などのための定数をプログラム全体の冒頭でつぎのように定義する。

1. **Const MAX As Integer = 9999**
2. **Const NIL As Integer = 0**
3. **Const HEAD = 1** ' リストの番兵 Head 1 行目
4. **Const CVAL = 1** ' リストのセルの値 A 列
5. **Const NEXTP = 2** ' リストのセルのポインタ B 列

- 未使用の行番号を見つける リストに新しいデータを追加するときには、未使用の行を見つける必要がある。そのために、Range オブジェクトの End プロパティを利用する。例えば、A 列でデータが格納されている最後の行は、つぎのようにして得られる。

```
Cells(1, 1).End(xlDown).Row
```

これは、A 列を 1 行目から下へ向かって調べていき、最初に見つかった空欄の直前の行番号を値とする。すなわち、図 10.6 の場合、空欄は 4 行目であることから 3 となる (5 行目以降の空欄は調べられない)。これに対し、例えば、10 行目から上に向かって調べるときには `Cells(10, 1).End(xlDown).Row` とする。この場合には 6 が得られる (空欄が 7 行目なので直前の行)。

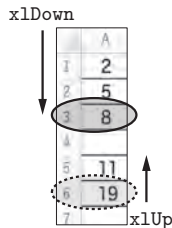


図 10.6 空欄の探索

イミディエイト ウィンドウ

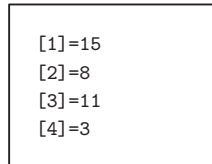


図 10.7 リストの出力例

これと対比される探索法に幅優先探索 (breadth-first-search) がある。この方法では、図 11.8(b) のように、根のつぎに深さ 1 のすべての頂点を、例えば、左から右にたどったあと、深さ 2 のすべての頂点、深さ 3 のすべての頂点とたどっていく。この結果、「a → b → c → d → e → f」の順で探索される。

深さ優先探索と幅優先探索は、対戦型ゲーム (五目並べやトランプなど) やパズルゲームなどを実装する際に有用である。これら探索法は、スタック、キュー (10.3 節, 10.4 節参照) を利用することで実装できることから、11.3.2 項以降では迷路の探索を例に取りあげながらその実装法を述べる。

なお、ここでは図 11.8 の 2 分木を用いて説明したが、本節で述べる探索法は一般の木にもあてはまる。

### 11.3.2 迷路の探索の実装法

図 11.9(a) の迷路を例に取りあげ、入口から出口 (2 か所) に至る経路を、深さ優先探索と幅優先探索のそれぞれで見つける方法について述べる。そのために、この迷路を同図 (b) のように、間仕切りされた空間 (区画) を頂点とし、移動できる区画どうしを辺で結んだうえで、入口を根とした 2 分木で表すことにする<sup>†</sup>。

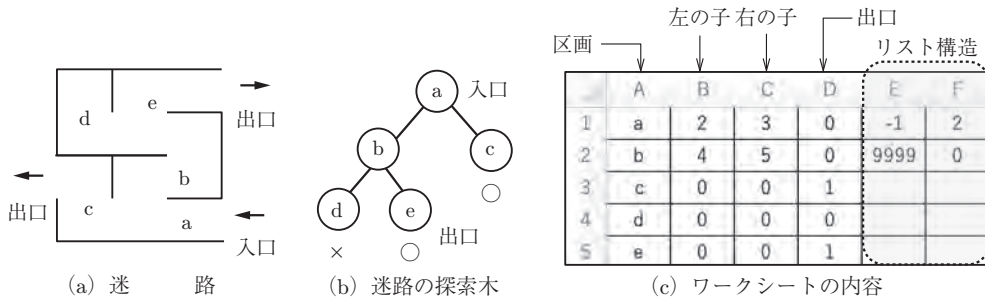


図 11.9 迷路の探索

これにより、入口 (根) から出口 (いずれかの葉) に至る経路 (通過する頂点の列) の探索は、図 11.9(b) の木の探索に帰着される。このような迷路をワークシート上に表現するために、同図 (c) のように各列を用いる。すなわち、A 列には頂点名 (区画名) を、B 列と C 列には 2 分木のポイントを、D 列には出口であれば「1」、そうでなければ「0」をそれぞれ代入する。さらに、E 列と F 列はリスト構造 (スタックまたはキュー) のために用いる。

実装にあたっては、つぎの Const 変数をプログラムのなかで用いる。

1. **Const MAX As Integer = 9999**
2. **Const NIL As Integer = 0**
3. **Const BTVAL = 1** ' 2 分木の値 A 列

<sup>†</sup> この迷路の場合は 2 分木として表されるが、一般的にはその限りではない。

```

4. Worksheets.Add.Name = "B" ' ワークシート B を新規生成(すでにあればこの行は不要)
5. Worksheets.Add.Name = "C" ' ワークシート C を新規生成(すでにあればこの行は不要)
6. Worksheets("CSV-test").Activate ' 入力するワークシートを選択
7. Ai = 1: Bi = 1: Ci = 1
8. For j = 1 To n ' ワークシート CSV-test の行番号 (1,2,3,...,n)
9.   If (Val(Cells(j, 4).Value) >= score) Then
10.    tmp = Range("A1").CurrentRegion.Rows(j).Value ' j 行目のレコード
11.    Select Case Cells(j, 3).Value ' 学部 (C 列) で選別
12.      Case "A" ' A 学部の場合
13.        Worksheets("A").Range("A1:D100").Rows(Ai).Value = tmp
14.        Ai = Ai + 1 ' ワークシート A の行の更新
15.      Case "B" ' B 学部の場合
16.        Worksheets("B").Range("A1:D100").Rows(Bi).Value = tmp
17.        Bi = Bi + 1 ' ワークシート B の行の更新
18.      Case "C" ' C 学部の場合
19.        Worksheets("C").Range("A1:D100").Rows(Ci).Value = tmp
20.        Ci = Ci + 1 ' ワークシート C の行の更新
21.    Case Else
22.    End Select
23.  End If
24. Next
25. End Sub

```

3~5 行目で三つのワークシートを新規作成している (すでに同名のワークシートがあればエラーになるので、これらの行は削除する)。9 行目で  $j$  行目の得点 (D 列) と `score` を比較し、`score` 以上のレコードのみを、11~22 行目で学部 (C 列) に応じたワークシートへ格納する<sup>†</sup>。このプログラムを `Distribute_faculty(128, 60)` として実行すれば、図 12.10 のように「得点が 60 以上のレコード」が学部別のワークシートに分配される。

#### 例 12.4 学部ごとの学籍番号の生成

各学部の学籍番号をつぎのように定め、学部ごとのワークシートの A 列に格納する。

A 学部：10202501~,      B 学部：20202501~,      C 学部：30202501~

そのために、`Distribute_faculty()` の 13 行目のあとに学籍番号を A 列に代入する行 (つぎの 14 行目) を追加する。これにより、ワークシート A を生成する際にこの学籍番号を A 列に代入することができる。ここで、 $A_i$  は学籍番号の連番 (1, 2, ...) である。

```

13. Worksheets("A").Range("A1:D100").Rows(Ai).Value = tmp
14. Worksheets("A").Cells(Ai, 1) = 10202500 + Ai ' 追加行

```

<sup>†</sup> `Val(Cells(j, 4).Value)` によりセルの値が数値に変換される (macOS の場合に効果的。例 12.1 参照)。

# 索引

<b>【あ】</b>	
後戻り	121
アプリケーション	10
アプリケーションソフトウェア	10
アルゴリズム	11, 73
<b>【い】</b>	
イベント処理	153
イミディエイト ウィンドウ	28, 40
インスタンス	15
<b>【う】</b>	
ウェブスクレイピング	144
ウォッチ ウィンドウ	24
<b>【え】</b>	
エディタ	13, 17
<b>【お】</b>	
オブジェクト	14
オブジェクト指向	14
オブジェクト指向プログラミング	14
オープンデータ	147
オペレーティングシステム	10
親	115
<b>【か】</b>	
開アドレス法	90
外部記憶	2, 102
外部整列	102
拡散モデル	155
型	45
合併	137
仮引数	38
関係	137
関係代数	137
関係データベース	137
関数	38, 44, 53, 69
関数呼び出し	38

<b>【き】</b>	
キー	83
木	115
記憶装置	2
基盤モデル	155
キュー	105, 110, 114, 125
共通部分	137
局所変数	70
<b>【く】</b>	
クイックソート	98
空リスト	105
組	137
クラウドサービス	145
クラス	15
繰り返し	58
<b>【け】</b>	
計算量	76
継承	15
結合	137
決定木	151
言語	1
言語処理系	1
<b>【こ】</b>	
子	115
交換	95
高水準言語	1
構造化データ	143
構文規則	1
コレクション	30
コンパイラ	11
コンパイラディレクティブ	93
<b>【さ】</b>	
差	137
最悪計算量	76
最下位ビット	5
再帰関数	68
再帰呼び出し	68
最上位ビット	5

サイズ	76
最大公約数	73
サンプリング周期	8
<b>【し】</b>	
時間計算量	76, 78
時間量	76
軸要素	98
指数時間	77
自然言語処理	155
子孫	115
実行可能プログラム	1
実行制御	58
実行停止	21
実行ファイル	1
実引数	38, 55
質問	137
射影	137
主記憶	2, 102
条件分岐	58
衝突	90
深層学習	150
<b>【す】</b>	
数学的帰納法	71
数値関数	38
スケーリング則	155
スタック	105, 108, 109, 114, 123
ステップ実行	25
ステートメント	44
<b>【せ】</b>	
正規化表現	5
制御構文	59
生成過程	156
生成 AI	155
整列	93, 120, 132
セル	2, 17, 30
線形繰り返しプロセス	81
線形再帰プロセス	68
線形時間	77
線形探索	84, 112, 138, 150
線形リスト	106



<b>【ろ】</b>		<b>【わ】</b>	
ローカル ウィンドウ	26	ワークシート	17, 30
◇			
<b>【A】</b>		<b>【G】</b>	
AI	150, 154	GB	5
ASCII コード	6		
<b>【B】</b>		<b>【I】</b>	
B (バイト)	5	IaaS	145
BASIC	18	IDE	14
Boolean	45	If	61
Boolean型	50	Integer	45
		Integer型	49
<b>【C】</b>		IoT	144
Call	69		
Cells	33	<b>【K】</b>	
Close 文	129	KB	5
Const ステートメント	107		
CPU	2	<b>【L】</b>	
CSV	127	LIFO	105
		Line 文	129
<b>【D】</b>		LLM	155
dequeue	105, 111	Long	45
Dim 宣言	46	Loop While	66
Double	45		
Double型	49	<b>【M】</b>	
Do While	66	MB	5
dpi	6		
<b>【E】</b>		<b>【N】</b>	
ELIZA	152	NIL	106, 116
END	59		
enqueue	105	<b>【O】</b>	
EOF	127	Open 文	128
Exit Do	67	Option ステートメント	44
Exit For	67	Option Explicit	48
Exit Function	67	OS	10
Exit Sub	67	$O(f(n))$	77
		$O(\log n)$	77
<b>【F】</b>		$O(n)$	77
False	50	$O(1)$	77, 91
FIFO	105		
For	63	<b>【P】</b>	
fps	7	PaaS	145
Function	44, 53, 69	pop	105, 109
		push	105
		<b>【R】</b>	
		Randomize	55
		Range	30
		RGB	7
		<b>【S】</b>	
		SaaS	145
		Select	62
		Single	45
		START	59
		String	45
		String型	51
		Sub プロシージャ	44, 69
		<b>【T】</b>	
		True	50
		<b>【V】</b>	
		Variant	45, 47
		Variety	143
		VBA	1, 10, 16, 44, 148
		VBE	10, 14, 17
		Velocity	143
		Volume	143
		<b>【W】</b>	
		Web API	148
		While	65
		Workbooks	30
		Worksheets	30
		Write # 文	131
		<b>【数字】</b>	
		2 進法	4, 9
		2 の補数表現	5
		2 分探索	85, 121
		2 分木	115, 150
		8 進法	9
		16 進法	9
		<b>【記号】</b>	
		#Const	94
		&	51
		'	45
		:	42, 45
		;	42
		?	32, 41

—— 著者略歴 ——

猪股 俊光 (いのまた としみつ)

1982年 八戸工業高等専門学校電気工学科卒業  
1984年 豊橋技術科学大学工学部生産システム工学課程卒業  
1986年 豊橋技術科学大学大学院工学研究科修士課程修了(生産システム工学専攻)  
1989年 豊橋技術科学大学大学院工学研究科博士後期課程修了(システム情報工学専攻), 工学博士  
豊橋技術科学大学助手  
1992年 静岡理工科大学講師  
1995年 静岡理工科大学助教授  
1998年 岩手県立大学助教授  
2007年 岩手県立大学教授  
現在に至る  
2007年 八戸工業高等専門学校非常勤講師(兼務)

呉 書雅 (う しゅうや)

1999年 (台湾) 高雄市立左营高級中学卒業  
2004年 (台湾) 中国文化大学国際貿易学科日本語文学科副専攻卒業  
2007年 (台湾) 中国文化大学大学院日本語文学研究科修士課程修了  
2010年 広島大学高等教育研究開発センター研究生修了  
2013年 広島大学高等教育研究開発センター研究員  
2016年 広島大学大学院教育学研究科博士後期課程単位取得満期退学(教育人間科学専攻)  
弘前大学非常勤講師  
2019年 東北労災看護専門学校非常勤講師  
2020年 東北大学大学院教育学研究科博士後期課程修了(総合教育科学専攻), 博士(教育学)  
東北大学学術研究員  
東北福祉大学非常勤講師(兼務)  
福島大学特任准教授  
2023年 岩手県立大学准教授  
現在に至る

## アルゴリズムとデータ構造を Excel VBA で学ぶ

Introduction to Algorithms and Data Structures Using Excel VBA

© Toshimitsu Inomata, Shuya Wu 2025

2025年9月30日 初版第1刷発行

★

検印省略

著者 猪股俊光  
呉書雅  
発行者 株式会社 コロナ社  
代表者 牛来真也  
印刷所 三美印刷株式会社  
製本所 有限会社 愛千製本所

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社  
CORONA PUBLISHING CO., LTD.  
Tokyo Japan

振替 00140-8-14844・電話(03)3941-3131(代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-02952-9 C3055 Printed in Japan

(田中)



JCOPY <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構(電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp)の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。