



電子情報通信レクチャーシリーズ B-8

データ構造とアルゴリズム

電子情報通信学会◎編

岩沼宏治

美濃英俊 共著

鍋島英知

山本泰生

コロナ社

▶電子情報通信学会 教科書委員会 企画委員会◀

- 委員長 ————— 原 島 博 (東京大学名誉教授)
- 幹事 ————— 石 塚 満 (東京大学名誉教授)
(五十音順)
- 大石進一 (早稲田大学教授)
- 中川正雄 (慶應義塾大学名誉教授)
- 古屋一仁 (東京工業大学名誉教授)

▶電子情報通信学会 教科書委員会◀

- 委員長 ————— 辻 井 重 男 (東京工業大学名誉教授)
- 副委員長 ————— 神 谷 武 志 (東京大学名誉教授)
- 宮 原 秀 夫 (大阪大学名誉教授)
- 幹事長兼企画委員長 ——— 原 島 博 (東京大学名誉教授)
- 幹事 ————— 石 塚 満 (東京大学名誉教授)
(五十音順)
- 大石進一 (早稲田大学教授)
- 中川正雄 (慶應義塾大学名誉教授)
- 古屋一仁 (東京工業大学名誉教授)
- 委員 ————— 122名

(2017年12月現在)

刊行のことば

新世紀の開幕を控えた 1990 年代、本学会が対象とする学問と技術の広がりとお行きは飛躍的に拡大し、電子情報通信技術とほぼ同義語としての“IT”が連日、新聞紙面を賑わすようになった。

いわゆる IT 革命に対する感度は人により様々であるとしても、IT が経済、行政、教育、文化、医療、福祉、環境など社会全般のインフラストラクチャとなり、グローバルなスケールで文明の構造と人々の心のありさまを変えつつあることは間違いない。

また、政府が IT と並ぶ科学技術政策の重点として掲げるナノテクノロジーやバイオテクノロジーも本学会が直接、あるいは間接に対象とするフロンティアである。例えば工学にとつて、これまで教養的色彩の強かった量子力学は、今やナノテクノロジーや量子コンピュータの研究開発に不可欠な実学的手法となった。

こうした技術と人間・社会とのかかわりの深まりや学術の広がりを踏まえて、本学会は 1999 年、教科書委員会を発足させ、約 2 年間をかけて新しい教科書シリーズの構想を練り、高専、大学学部学生、及び大学院学生を主な対象として、共通、基礎、基盤、展開の諸段階からなる 60 余冊の教科書を刊行することとした。

分野の広がりに加えて、ビジュアルな説明に重点をおいて理解を深めるよう配慮したのも本シリーズの特長である。しかし、受身的な読み方だけでは、書かれた内容を活用することはできない。“分かる”とは、自分なりの論理で対象を再構築することである。研究開発の将来を担う学生諸君には是非そのような積極的な読み方をしていただきたい。

さて、IT 社会が目指す人類の普遍的価値は何かと改めて問われれば、それは、安定性とのバランスが保たれる中での自由の拡大ではないだろうか。

哲学者ヘーゲルは、“世界史とは、人間の自由の意識の進歩のことであり、… その進歩の必然性を我々は認識しなければならない”と歴史哲学講義で述べている。“自由”には利便性の向上や自己決定・選択幅の拡大など多様な意味が込められよう。電子情報通信技術による自由の拡大は、様々な矛盾や相克あるいは摩擦を引き起こすことも事実であるが、それらのマイナス面を最小化しつつ、我々はヘーゲルの時代的、地域的制約を超えて、人々の幸福感を高めるような自由の拡大を目指したいものである。

学生諸君が、そのような夢と気概をもって勉学し、将来、各自の才能を十分に発揮して活躍していただくための知的資産として本教科書シリーズが役立つことを執筆者らと共に願っ

ii 刊 行 の こ と ば

ている。

なお、昭和 55 年以来発刊してきた電子情報通信学会大学シリーズも、現代的価値を持ち続けているので、本シリーズとあわせ、利用していただければ幸いである。

終わりに本シリーズの発刊にご協力いただいた多くの方々に深い感謝の意を表しておきたい。

2002 年 3 月

電子情報通信学会 教科書委員会
委員長 辻 井 重 男

まえがき

本書で取り扱うデータ構造とアルゴリズムに関する研究は、コンピュータ科学の挑戦の歴史の中で、常に中心の位置を占めてきた。そこで開発された技法なくしては、現在の高度情報化社会の実現はあり得なかったことに疑問をはさむ余地はない。そのため専門教育における初学者用の教科書はとても大事であり、既に多くの成書が出版されている。本書を執筆するにあたって、類書の中でどのような特徴を出すかは大きく悩むところであった。本書の執筆者4人は地方国立大学において実際に関連の専門教育に長年携わってきた人間である。いろいろと議論を行った結果、以下のような執筆方針が徐々に固まっていった。

- 初学者用であるので、取り上げるトピックは奇をてらわずに、正攻法で選択する。
- 本シリーズ全体の要請にもなっているが、説明は図を多用して平易に行う。数式の使用はなるべく控えて、使用する場合でも2行以上は連続させないことを原則とした。また解説するトピックは天下りの説明は避け、取り上げる理由や、結果を支えるアイデアから説明する。初学者には不要なレベルの厳密な説明や証明は行わない。
- 説明を平易に行うからといって、表面的なものだけに流れずに本質はきちんと説明する。また学習者が自ら興味をもってより進んだ勉学に取り組むように、裏に隠れた深い性質などについても一部、光を当てるように工夫する。

以上は初学者用教科書での一般的な方針であるが、“データ構造とアルゴリズム”の分野固有の観点からは以下の方針を考えた。

- 昨今のメモリの大容量化に代表されるハードウェアの高性能化と、高速ネットワークの普及と機械学習技術の発展に伴うデータの巨大化などの動向に留意し、その解決につながる技術を積極的に取り上げ、解説を行う。例えば、メモリ(領域)消費型のデータ構造とその上のアルゴリズム技法を、類似の成書より積極的に取り上げる。
- 昨今のシステムの大規模化と高信頼化に対応できる人材を育成するためには、ソフトウェアのカプセル化や抽象化、一般化、検証などの技術を初学者のレベルから教えることが重要である。ソフトウェア工学とは別の次元で、プログラミングレベルでの知識とスキルを身に付けさせるために、オブジェクト指向型言語であるC++による実装コードの教示を行う。
- 本書の前半ではC++言語での実装コードを示すが、これは初学者によいコードを読ませることが目的であり、Art of Programmingにつながるような解説を付記するよう

に努力する。後半では疑似コードでアルゴリズムを示し、論理的かつ抽象的な思考能力の育成を図る。

オブジェクト指向型の C++ 言語は、初学者教育での利用には種々の議論もあるが、早いうちから、カプセル化や継承、多態性 (polymorphism) などの概念に親しませることは、やはり高度な専門性の修得を目指す読者には有用であると考ええる。また STL などの汎用の高性能ライブラリが利用できることも利点である。1.2 節には、C 言語などは知っているが、C++ 言語はよく知らない読者のための導入解説を行って、敷居を下げる工夫を行っている。

疑似コードレベルのアルゴリズム記述や問題の特性把握は、抽象的かつ論理的な思考、あるいは一般化した思考の訓練になり、とても有用である。昨今ではネット上には著名なアルゴリズムの実装コードが多数公開されている。ややもすると、それらを表面的にだけ参考にし、本質的な洞察を行わずに済ませてしまう例が数多く見受けられる。そのためか、アルゴリズムの最適化や改良のために疑似コードでの記述と整理を行わせると、ほとんどできない人が思いのほか多いことに驚かされる。そもそも疑似コードはプログラムの設計図となるものであり、上級仕様の作成などを行うためにも必要なスキルであるので、是非、多くの読者に習熟してほしいと考えている。

本書は1年間もしくは1年半程度の勉強期間を想定して^{へんさん}編纂されている。執筆陣は本書の前半を、専門技術者教育におけるプログラミング言語教育の後半(1年後期)の教材として利用することを想定している。本書の後半は、学部2年から開始されるデータ構造とアルゴリズムの講義演習に使用することを想定している。

昨今では、ハードウェアの高性能化と高機能なソフトウェア統合開発環境の普及に伴い、アルゴリズムとデータ構造に関する専門的な勉強をしなくとも、基本的で小規模なソフトウェアならば容易に開発できる時代になっている。これは「誰でもプログラマになれる」という意味では望ましいことであるが、情報処理技術の本質に関して誤解が生じやすいという意味では望ましくないと考えられる。より先進的で高度なシステムを設計し開発するためにも、あるいは大規模かつ高信頼性を必要とするシステムを選定し運用するためにも、アルゴリズムとデータ構造に関する専門的な知識とスキルは必要不可欠なものである。より多くの読者に本分野を学んでいただき、現在および将来の高度情報化社会を支えていただければ、幸いと考える。本書がその一助になれば、執筆者らの望外の喜びである。

最後に、遅々として進まぬ執筆作業を忍耐強く待ち、励ましをいただいたコロナ社の皆様に感謝を申し上げます。次第です。

2017年12月

著者を代表して 岩 沼 宏 治

目 次

1. はじめに

1.1	天文学的数字とコンピュータ科学的数字はどちらが大きいか?	2
1.2	データ構造のプログラム表現：オブジェクトとクラス	4
1.2.1	クラスとオブジェクト	5
1.2.2	値渡しと参照渡し	8
	本章のまとめ	9
	理解度の確認	10

2. データ構造の基礎

2.1	計算とメモリ	12
2.2	配 列	13
2.2.1	固定長配列	14
2.2.2	可変長配列	15
談話室	C++ 標準テンプレートライブラリの <code>vector</code> クラス	18
2.3	連結リスト	19
談話室	C++ 標準テンプレートライブラリの <code>list</code> クラス	26
2.4	スタックとキュー	27
2.5	木 構 造	32
	本章のまとめ	40
	理解度の確認	40

3. 基本的な探索整列の手法

3.1	アルゴリズムと計算量	42
3.2	素朴な探索	46

談話室 任意のキーによる探索	48
3.3 再帰的探索	49
3.4 素朴な整列	52
3.4.1 選択ソート	52
3.4.2 挿入ソート	53
3.5 再帰的整列	56
3.5.1 マージソート	56
3.5.2 クイックソート	59
3.6 空間を利用する整列	63
3.6.1 パケットソート	63
3.6.2 計数ソート	65
3.6.3 基数ソート	66
本章のまとめ	69
理解度の確認	70

4. 二分木とその応用

4.1 二分探索木	72
4.1.1 素朴な二分探索木	72
談話室 多態性	74
4.1.2 平衡二分探索木	77
談話室 さまざまな平衡木	82
4.2 優先度付きキューとヒープソート	84
4.2.1 優先度付きキュー	84
4.2.2 ヒープによる高速な優先度付きキュー	84
4.2.3 ヒープソート	88
4.3 最近傍探索と kd-木	90
4.3.1 最近傍探索	91
4.3.2 kd-木の構築	94
本章のまとめ	95
理解度の確認	95

5. ハッシュ表

5.1	ハッシュ表の原理	98
5.1.1	分離チェーン法	99
5.1.2	ハッシュ関数の設計	100
5.1.3	文字列キーに対するハッシュ関数	101
5.2	開番地法	103
5.2.1	開番地法の原理	103
5.2.2	開番地法における削除	106
5.2.3	代替アドレス	107
5.2.4	ハッシュ表の拡大	108
	談話室 ハッシュ関数と認証	109
	本章のまとめ	110
	理解度の確認	110

6. グラフ

6.1	グラフの表現と探索	112
6.2	最小全域木問題	116
6.3	最短経路問題	122
6.3.1	単一始点問題：ダイクストラ法とベルマン・フォード法	123
6.3.2	単一点対問題：A* アルゴリズム	127
6.4	最長経路問題：トポロジカルソート	132
	本章のまとめ	136
	理解度の確認	136

7. 文字列照合

7.1	文字列照合問題と素朴な解法	138
7.1.1	力まかせ法	139
7.2	高速な文字列照合法	140
7.2.1	力まかせ法の欠点と BM 法の原理	141

7.2.2	BM 法の実際	142
談話室	BM 法の補足	147
7.3	ハッシュ法を用いた文字列検索	149
7.4	索引に基づく高速文字列照合	150
7.4.1	トライに基づく文字列照合	151
7.4.2	パトリシアトライによる文字列照合	154
7.4.3	サフィックス木に基づく文字列照合	156
	本章のまとめ	159
	理解度の確認	160

8. アルゴリズム技法

8.1	分割統治法	162
8.1.1	分割統治法の適用例	163
8.1.2	分割統治法の性質	166
談話室	シュトラッセン (Strassen) のアルゴリズム	170
8.2	動的計画法	170
8.2.1	0-1 ナップサック問題と動的計画法	172
8.3	分枝限定法	176
8.3.1	緩和法と上界見積り	179
8.4	オンライン近似計算：ストリームマイニング	184
8.4.1	ストリームマイニング	184
8.4.2	オンライン近似計算と Space Saving 法	186
談話室	オンライン計算と近似計算の枠組みについて	190
	本章のまとめ	191
	理解度の確認	191

引用・参考文献	193
---------	-----

索引	194
----	-----

1

はじめに

本章では 2 章からの本論に先立ち、まず初めに天文学的数字とコンピュータ科学的数字という概念の比較を行い、それを通して「データ構造とアルゴリズム」を勉強することの必要性や重要性を深く理解してもらうことを試みる。次に後半では、2 章以降の C++ 言語の実行コードをよく理解するために、オブジェクトとクラスについて簡潔な解説を行う。「C 言語などを既に知っているが C++ 言語は初めて」という読者は多いと思う。コンパクトにまとめているので、是非、精読してみてほしい。

1.1

天文学的数字とコンピュータ科学的数字はどちらが大きいか？

天文学は古くから世界各地で発達している学問である。最も古い地域では4000年以上の歴史があり、人々の生活にも深く浸透している。このため“天文学的数字”や“星の数ほどある”という表現は、世界中で普遍的に使われている。一方のコンピュータ科学 (computer science) は、1920年代の数理論理学における計算可能性の研究に発端を有するが、第2次世界大戦後の1950年代後半から本格的な発展を遂げた非常に若い学問である。残念ながら、コンピュータ科学はまだ世間一般の人にはよく理解されておらず、“コンピュータ科学的数字”などという表現は普段の生活では全く使われていない。そこで本節では、この“コンピュータ科学的数字”というものを新しく考えて、“天文学的数字”との比較を通して、コンピュータ科学がこれまでいかに困難な課題に取り組んできた重要な学問であるかを、読者各位に理解してもらおうと思う。

まず、天文学的数字とは非常に巨大な数を表現したものであるが、それはどの程度の数であろうか？ その明確な定義はどこにもない。そこで類似の表現である“星の数ほどある”という表現に置き換えて考えてみる。星の数もかなり曖昧な概念であるが、現代物理学では宇宙の大きさなどが推定されているので、それを基本として、ある程度の合理性をもって見積もれると思われる。現代物理学においても宇宙の形状や大きさにはさまざまな議論がある。宇宙論は筆者の専門外であり、またここでは、一般の人々が抱くイメージにおける星の数を見積もるので、宇宙の形状も最も想像しやすい“球状の宇宙”を仮定することにする。その直径は最新の理論と観測によって470億光年程度と見積もられているようなので、これを採用してメートル表示してみる。光の速度は秒速約30万kmとされており、メートル表示すると秒速 3.0×10^8 mとなる。よって1光年、即ち光が1年間に進む距離は約 9.5×10^{15} ($\approx 3.0 \times 10^8 \times 60 \times 60 \times 24 \times 365$) mとなる。よって470億光年は約 4.5×10^{26} mとなる。宇宙の半径はおおよそ 10^{27} m程度以下と思ってよいことになるので、これに基づき宇宙の体積を計算する。よく知られた球体積の公式より、 $4\pi/3 \times (10^{27})^3 \text{ m}^3$ となるので、宇宙の体積を概数として 10^{82} m^3 程度と考えることにする。

以上を前提として星の数の推測を試みよう。星の定義もさまざまあるが、ほとんどの一般の人が直感的に想像する“星”なるものは、多くとも 1 m^3 当り1個程度であり、それ以上は

存在しないと考えられる。この前提に立てば、星の数は多くとも 10^{82} 個程度と推定できることになり、これが天文学的数字の一つの具体的な値となる。別の観点からは、我々がいる銀河系には数千億個の恒星があり、同種の銀河は（観測できる）宇宙の中に少なくとも 1700 億個程度はあると説明されている。この説明に従えば、1 兆は 10^{12} であるので、恒星は宇宙全体で 10^{24} 個程度はあるという結論になる。これを考慮したとき、 10^{82} はかなり余裕のある値であり、星の数の上限としてはかなり堅い推定値になっている。

さて、これに対して“コンピュータ科学的数字”とはどの程度の大きさの数を指すのであろうか？ コンピュータ科学的数字も明確な概念ではなく、その定義はどこにもない。コンピュータ科学の最も輝かしい成果物の一つが最先端の CPU プロセッサであることに、疑念をささむ余地はないであろう。この CPU を外部から見た場合、CPU は入出力端子上の信号を制御する装置と見なすことができる。現在普及しているマルチコア型 CPU は 10 億を超えるトランジスタからなる非常に複雑な集積回路であるが、入出力もメモリアクセスの高速化のためにマルチチャネル化しており、入出力端子数も 2000 を超えるのが普通である。より基本的なシングルコア型 CPU でも、入出力端子の本数が 300 を超えるものがあり、その場合の入出力信号の総数は単純計算では 2^{300} 程度の巨大な値となる。そこで、このシングルコア型 CPU の入出力信号の総数をコンピュータ科学的数字の一つの具体的な数値と考えることにする。CPU はコンピュータ科学・工学の象徴であり、また入出力信号の総数は CPU 設計の複雑さあるいは困難さを表す指標となることも考え合わせると、これは妥当な設定であると思われる。

以上の考察を基に、コンピュータ科学的数字 2^{300} と天文学的数字 10^{82} を比較してみる。まず容易に以下のことがわかる。

$$2^{300} = (2^{10})^{30} = (1.024 \times 10^3)^{30} > 10^{90}$$

よって、 $2^{300} > 10^{82}$ であることは明らかである。多くの仮定には基づいているが、一つの結論として「コンピュータ科学的数字のほうが天文学的数字よりもはるかに大きい」ということができる。

より重要なことは、日常生活における“天文学的数字”とか“星の数ほどある”という表現は、否定的な文脈の中で使われる場合がほとんどであることである。“目標の達成が不可能なほどに大量にある、あるいは巨大である”という表現とほとんど同義な語として認識、使用されている。これに対して、コンピュータ科学は同程度以上の巨大なデータあるいは複雑さに正面から挑戦し、克服してきた歴史をもつ学問である。我々が現在利用している高性能 CPU や高速ネットワーク、高機能ソフトウェアなどはその輝かしい成果物である。

上記の二つの数は、現在の最高速コンピュータの演算能力と比較しても格段に巨大である

ことに注意してほしい。現在の最高速 CPU の基本動作速度はたかだか 10 GHz 程度[†]であり、また並列に同時動作する CPU コアもたかだか 10 万個程度である。よって 1 秒間に行える基本命令の数は 10^{15} ($= (10 \times 10^9) \times 10^5$) 程度であり、1 年間に実行できる基本命令は 3.2×10^{22} 個程度にとどまる。したがって、宇宙の年齢にも匹敵すると推定される 100 億年の時間をかけても、実行できる基本命令の数は 3.2×10^{32} 個程度にしかならない。コンピュータ科学的数字あるいは天文学的数字のほうがはるかに大きいことを、ぜひ理解してほしい。

このように実際に手に入る物理的計算パワーと、解くべき問題の複雑さの間には、常に大きなギャップがある。それを埋めてきたのが、本書で取り扱うアルゴリズムとデータ構造に関する技法である。アルゴリズムとデータ構造に関する研究は、コンピュータ科学の挑戦の歴史の中で、常に中心の位置を占めてきた。そこで開発された技術なくしては、現在の高度情報化社会の実現はあり得なかったと考えられる。現在のビッグデータ処理、機械学習、自然言語処理、コンピュータグラフィックス、画像音声処理、インターネットあるいは高性能なソフトウェア統合開発環境など、すべてのコンピュータ関連技術の基盤を支えている。アルゴリズムとデータ構造に関する知識とスキルは、次世代に求められるより先進的で高機能なシステムを開発するためにも、あるいは現在、実際に求められる大規模かつ高信頼性を必要とするシステムを適切に選定し運用するためにも、必要不可欠なものとなっている。

1.2 データ構造のプログラム表現：オブジェクトとクラス

アルゴリズムとデータ構造に関する教科書は数多く存在するが、その多くは C 言語や Pascal 風の疑似言語などの手続き型言語を用いて記述されている。手続き型言語は単純でわかりやすく小規模なプログラムの作成に向いているが、大規模なプログラムの作成や開発・保守のしやすさに関してはオブジェクト指向言語のほうが適している。実社会におけるプログラミングにおいても、オブジェクト指向言語が幅広く利用されるようになって久しい。

本書の前半ではアルゴリズムとデータ構造をオブジェクト指向言語の代表格の一つである C++ により記述する。これにより、基本的なアルゴリズムやデータ構造をオブジェクト指向言語で表現する手法を学ぶ。また実際に動作する具体的なプログラム例を示すことにより、

[†] クロック周波数が 10 GHz ということは、1 クロックは 100 億分の 1 秒となり、光でさえ 3 cm しか進めない微小時間である。

初学者の理解を助ける狙いもある。

本書後半の応用的なアルゴリズムとデータ構造については、その記述にしばしば Pascal 風の疑似コードを用いる。複雑な処理を計算機に実行させたい場合、いきなりプログラムを組み始めるのではなく、まずは処理の手順を論理的かつ明解に表現することが非常に重要である。疑似コードはこの目的のためによく使われる道具であり、処理の本質的な部分を簡潔に表現することが可能である。適切な疑似コードが与えられたならば、それを C++ や他のプログラミング言語に翻訳し直すことは難しいことではない。

本節では、本書に記載のプログラムを読むにあたって必要な C++ の基礎知識を解説する。ただし本書はプログラミング言語の入門書ではなく、また紙面の都合もあるため、C 言語の知識を前提として、C++ との差異に焦点を当てて簡単に解説する。また必要に応じて次章以降でも C++ の解説を行う。C++ の詳細については C++ の教科書などを参照されたい（例えば文献 1）など。もし読者がすでに C++ に習熟している場合は本節を読み飛ばしてもらって構わない。

1.2.1

クラスとオブジェクト

C++ は基本的に C 言語の上位互換であり、if や for 文などの制御構文や、int や double などのデータ型、関数の定義や呼び出しなどは、C++ でも同様に利用できる。C++ と C の最も大きな違いは、C++ ではクラスが利用可能な点にある。以下ではクラスについて簡単に紹介する。

C++ では、C と同様に、整数を表す int 型や、倍精度浮動小数点数を表す double 型、文字を表す char 型などが利用可能であるが、これらの基本的な型だけでは十分であるとはいえない。例えば学生データや商品データなどを表す型があれば、学生データを成績順に整列したり、特定の商品の在庫数を調べるプログラムが、基本型だけで組むよりも、より簡潔に記述できる。

C++ ではクラスを利用して新しい型を定義できる。リスト 1.1 は複素数クラス `Complex` の定義例である。5 行目の `class Complex {` から 25 行目までがクラスの定義である。複素数は実部と虚部を表す二つの実数からなる。ここでは、double 型の変数 `re`, `im` を、それぞれ実部と虚部を表す変数として用意している (7, 8 行目)。このように“新しい型”は、いくつかの“既存の型”を組み合わせで定義する。“既存の型”には、int や double などの C++ が提供する型だけでなく、ユーザが定義した型も利用できる。変数 `re`, `im` のように、新しい型が内部に保持する変数をメンバ変数という。クラスによって定義された“新しい型”のデータをオブジェクトまたはインスタンスという。“新しい型”のオブジェクトを生

索

引

【あ】

間順35
アイテム184
アイテムストリーム185
値渡し8
アッカーマン関数の逆関数
.....121
後入れ先出し27
後順35
アルゴリズム42
アロー演算子23
安定な整列アルゴリズム55

【い】

一方向ハッシュ関数109
イベント頂点グラフ132
インスタンス5
インデックス138
イントロソート61

【え】

枝32
エンキュー30

【お】

オブジェクト5
オブジェクト指向言語4
重み112, 113
親クラス28
親頂点32, 116
オンラインアルゴリズム
.....55, 184

【か】

開番地法103
カウンタ187
片方向リスト25
カプセル化7
可変長配列クラス18
空の木32
完全二分木33, 77
緩和法179
緩和問題179

【き】

木32, 116
キー46
木構造32
基数ソート66
疑多項式時間アルゴリズム176
木の高さ119
キュー30
求解操作177
許容的130
競合比190
共通接頭辞検索152
近似アルゴリズム133, 184
近似度190

【く】

クイックセレクト94
クイックソート59
クヌース・モリス・プラット法
.....147
組合せ爆発173
クラス5
クラスカル法117
クラスプレート17
グラフ112
クリティカルパス132

【け】

計算量43
継承28
計数ソート65
経路114
経路の圧縮121
限定操作177
厳密計算185

【こ】

構造体7
項目46
子クラス28
子頂点32, 116
コンストラクタ6
コンピュータ科学2

【さ】

再帰計算169
再帰呼び出し38
最近点对163
最近傍探索91
最小スパニング木問題116
最小全域木116
最小全域木問題116
最大計算量44
最短経路問題122
最長共通部分文字列159
最長経路問題132
最長単純経路問題133
最適化問題172
先入れ先出し30
索引151
サフィックス木156
サフィックス配列157
参照渡し8
暫定解177

【し】

時間計算量43
辞書式順序49
指数時間46
実行形式12
実頻度188
始点113
終点113
充填率107
縮小法167
出現頻度185
シュトラッセン法170
準最適解133
順序対113
上界179, 186
衝突99
初期注目点139

【す】

スケジューリング問題132
スコープ演算子73
スタック27
ストリームマイニング184

スパース 154

【せ】

整数解ナップサック問題 .. 192
 整列 52
 接頭辞 151
 接尾辞 151
 接尾辞木 156
 全域性 116
 漸近の下界 45
 漸近の上界 45
 漸近の評価 45
 線形時間 45
 線形走査 107
 線形探索 47
 選択ソート 52

【そ】

疎 154
 走査 35
 相対最小頻度 185
 相対最大誤差 186
 挿入ソート 53
 双方向リスト 26
 ソート 52
 疎なグラフ 114

【た】

ダイクストラ法 123
 対数時間 45
 代替アドレス 103
 多項式時間 46
 タスク頂点グラフ 132
 多態性 74
 探索 46, 114
 単純経路 133

【ち】

力まかせ法 139
 逐次探索 47
 チャレンジ 109
 チャレンジレスポンス認証 109
 頂点 112

【て】

定数時間 45
 データストリーム 184
 テキスト 138
 デキュー 30
 デストラクタ 17
 手続き型言語 4
 天井関数 50

【と】

動的計画法 126, 170
 ドット演算子 7
 トポロジカル順序 133
 トポロジカルソート 133
 トライ 151
 トライ行列 152
 トランザクション 184
 貪欲法 117, 123, 127, 179

【な】

なぞり 35

【に】

二色木 82
 二分木 33
 二分探索 49
 二分探索木 72
 認証 109

【ね】

根 32, 116
 根付き木 32, 116, 119

【は】

葉 32, 116
 配列 13
 配列長 138
 バケット 64, 189
 バケットソート 63
 パターン 138
 パターン注目点 139
 ハッシュ関数 99
 ハッシュ値 99
 ハッシュ表 98, 189
 パトリシアトライ 155
 幅優先探索 35, 114
 番地 13

【ひ】

ヒープ 85
 ピボット 59, 162
 ヒューリスティック関数 .. 128
 ヒューリスティック近似解法
 127
 標準テンプレートライブラリ
 18
 頻出アイテム 185
 頻度誤差 188
 頻度サマリ 187

【ふ】

フィボナッチ数列 169

深さ 32
 深さ優先探索 35, 114
 負グラフ 133
 プッシュ 27
 負閉路 126
 分割統治法 162
 分枝限定法 176
 分枝操作 177
 分離チェーン法 99

【へ】

平均計算量 44
 併合 56
 平衡二分探索木 77
 平方走査 107
 並列処理 169
 閉路 114
 ベルマン・フォード法 126
 辺 112

【ほ】

ボイヤー・ムア法 140
 ポップ 27

【ま】

マージ 56
 マージソート 56, 162
 前順 35
 マスター定理 166
 末尾要素へのポインタ付き
 リスト 25

【み】

密なグラフ 114
 見積り頻度 188

【む】

向き 112
 無向グラフ 113
 無向辺 113

【め】

メモ化 172
 メモリリーク 17
 メンバ関数 6
 メンバ変数 5

【も】

文字コード 145
 文字列照合問題 138

【ゆ】

有向グラフ 113
 有向辺 113

優先度84
 優先度付きキュー84
 床関数50

【ら】

ラビン・カーブ法149
 乱択化61, 185
 ランダムアクセスメモリ13

【り】

リスト20
 領域計算量43
 領域探索91
 リングバッファ31
 隣接113
 隣接行列113
 隣接リスト114

【れ】

レコード46
 レスポンス109
 レベル順36
 連結116
 連結グラフ116
 連結リスト20
 連続ナップサック問題179

【A】

ASCII145
 assert マクロ16
 AVL 木77
 AVL バランス77
 A* アルゴリズム128

【B】

B 木82
 BDS trie154
 BM 法140

【F】

FIFO30

【K】

kd-木93
 KMP 法147

【L】

LC 法187

LCS159
 LIFO27
 list クラス26
 Lossy Counting 法187

【N】

n -分木33
 NP-困難133, 176

【O】

O 記法45

【R】

RAM13

【S】

Space Saving 法187
 SS 法187
 STL18

【U】

UNION-FIND 問題118

【V】

vector18

【記号】

:: (スコープ演算子)73

【ギリシャ文字】

ϵ -完全性188
 ϵ -劣近似性190
 Ω 記法45
 Θ 記法45

【数字】

0-1 ナップサック問題172
 2 重索引189
 2 進木トライ154

— 著者略歴 —

岩沼 宏治 (いわぬま こうじ)
1985年 東北大学大学院工学研究科博士前期課程修了 (電気及び通信工学専攻)
工学博士 (東北大学)
現在, 山梨大学教授

美濃 英俊 (みの ひでとし)
1989年 名古屋大学大学院理学研究科博士課程修了 (物理学専攻)
理学博士 (名古屋大学)
現在, 山梨大学教授

鍋島 英知 (なべしま ひでとも)
2001年 神戸大学大学院自然科学研究科博士課程修了 (情報メディア科学専攻)
博士 (工学) (神戸大学)
現在, 山梨大学准教授

山本 泰生 (やまもと よしたか)
2010年 総合研究大学院大学複合科学研究科博士課程修了 (情報学専攻)
博士 (情報学) (総合研究大学院大学)
現在, 山梨大学助教

データ構造とアルゴリズム

Data Structures and Algorithms

© 一般社団法人 電子情報通信学会 2018

2018年2月23日 初版第1刷発行

検印省略

編者 一般社団法人
電子情報通信学会
<http://www.ieice.org/>

著者 岩沼 宏治
美濃 英俊
鍋島 英知
山本 泰生

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 <http://www.coronasha.co.jp>

ISBN 978-4-339-01823-3 C3355 Printed in Japan



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。