

■コンピュータサイエンス教科書シリーズ **2**

データ構造とアルゴリズム

博士(工学) 伊藤 大雄 著



COMPUTER SCIENCE TEXTBOOK SERIES



コロナ社

コンピュータサイエンス教科書シリーズ編集委員会

編集委員長 曾和 将容（電気通信大学）

編集委員 岩田 彰（名古屋工業大学）

（五十音順）

富田 悦次（電気通信大学）

（2007年5月現在）

刊行のことば

インターネットやコンピュータなしでは一日も過ごせないサイバースペースの時代に突入している。また、日本の近隣諸国も IT 関連で急速に発展しつつあり、これらの人たちと手を携えて、つぎの時代を積極的に切り開く、本質を深く理解した人材を育てる必要に迫られている。一方では、少子化時代を迎え、大学などに入学する学生の気質も大きく変わりつつある。

以上の状況にかんがみ、わかりやすく体系化された、また質の高い IT 時代にふさわしい情報関連学科の教科書と、情報の専門家から見た文系や理工系学生を対象とした情報リテラシーの教科書を作ることを試みた。

本シリーズはつぎのような編集方針によって作られている。

- (1) 情報処理学会「コンピュータサイエンス教育カリキュラム」の報告、ACM Computing Curricula Recommendations を基本として、ネットワーク系の内容を充実し、現代にふさわしい内容にする。
- (2) 大学理工系学部情報系の 2 年から 3 年の学生を中心にして、高専などの情報と名の付くすべての専門学科はもちろんのこと、工学系学科に学ぶ学生が理解できるような内容にする。
- (3) コンピュータサイエンスの教科書シリーズであることを意識して、全体のハーモニーを大切にするとともに、単独の教科書としても使える内容とする。
- (4) 本シリーズでコンピュータサイエンスの教育を完遂できるようにする。ただし、巻数の制限から、プログラミング、データベース、ソフトウェア工学、画像情報処理、パターン認識、コンピュータグラフィックス、自然言語処理、論理設計、集積回路などの教科書を用意していない。これらはすでに出版されている他の著書を利用していただきたい。

ii 刊 行 の こ と ば

- (5) 本シリーズのうち「情報リテラシー」はその役割にかんがみ、情報系だけではなく文系、理工系など多様な専門の学生に、正しいコンピュータの知識を持ったうえでワープロなどのアプリケーションを使いこなし、なおかつ、プログラミングをしながらアプリケーションを使いこなせる学生を養成するための教科書として構成する。

本シリーズの執筆方針は以下のものである。

- (1) 最近の学生の気質をかんがみ、わかりやすく、丁寧に、体系的に表現する。ただし、内容のレベルを下げることはしない。
- (2) 基本原理を中心に体系的に記述し、現実社会との関連を明らかにすることにも配慮する。
- (3) 枝葉末節にとらわれてわかりにくくならないように考慮する。
- (4) 例題とその解答を章内に入れることによって理解を助ける。
- (5) 章末に演習問題を付けることによって理解を助ける。

本シリーズが、未来の情報社会を切り開いていけるたくましい学生を育てる一助となることができれば幸いです。

2006年5月

編集委員長 曾和 将容

まえがき

本書はデータ構造とアルゴリズムの重要部分について基礎から最新までを記した本である。対象は主に大学の学部2～3年の講義を想定しているが、大学院でも使用できる進んだ内容も含んでいる。

「データ構造とアルゴリズム」と題した書籍は数多く出版されており、それらの中には良書、好著が数多く含まれる^{†1}。それらがあるにもかかわらず私が本書を記したのにはもちろん意味がある。「データ構造とアルゴリズム」の世界は日進月歩であり、つぎからつぎへと新しく重要な技術が生まれてくる。そしてそのうちのいくつかは基礎的な必須のアルゴリズムとして残っていく。そういったアルゴリズムの中で非常に重要であるにもかかわらず、なまじ先進的であるために基礎的記述が中心となる教科書にまったく出現しないものが少なからず存在する。それらを知るためには現状では原著論文か、せいぜい英語の専門書に当たるしかない。しかしわれわれ専門家は別として、学生や一般企業のエンジニアにとって、そういった文献に当たれといわれても難があるだろう^{†2}。

本書はそれらを解決することを主眼として書かれている。したがって、類書にはない、最先端のアルゴリズムを解説することに多くのページを割いている。最先端のアルゴリズムはもちろん比較的難解である。しかし前述のように本書の主対象者である学部生でも理解できるよう、極力平易な解説を試みた。それらを完全に理解するには相当な努力が必要とされるかもしれないが、主たる考え方を理解することはさほど難しくないと考えている。

そして本書の最も重要な特長は、定数時間アルゴリズムの紹介に多くのペー

†1 参考文献リストに挙がっている教科書はどれもそうだが、数冊選ぶのなら、例えば文献(39), (43), (44), (50)などであろうか。

†2 実際、原著論文にはしばしば間違いが含まれており、その修復は専門家でない無理であろう。

ジを割いていることである。定数時間アルゴリズムとは、入力の中のほんの一部、定数個のデータだけを見て判定、計算するアルゴリズムの枠組みである。20年ほど前に現れ、今世紀になって急激に発展した。当初はほぼ理論的興味だけで研究が行われてきたが、最近ビッグデータとの絡みで実用面での研究も行われている^{†1}。いまや国際会議のメインピックの一つとなっているが、それを解説する専門書は非常に少ない。まして日本語で解説したものはほとんどなく、あるのは筆者やその共同研究者の書いた短めの解説^{(47), (48), (65)}ぐらいで、これだけきちんと解説した日本語の教科書は初めてとあってよかろう。定数時間アルゴリズムは21世紀のアルゴリズムの標準技術となることは間違いなく、これからアルゴリズムを学ぶ者はこの機会に是非とも学んでおいてほしい。

本書で使用する変数や関数などの表記法 (notation) について、その主なものを“記号表”という形で目次の後にまとめた。また、本書で使用する数学に関する基本的な用語および基本公式については、8章に簡単にまとめたので、適宜必要に応じて参照してほしい。さらに、各章末に演習問題を、また必要に応じてプログラム演習を掲載したので、本書の理解の助けになるだろう^{†2}。

本書の使い方についての一例を記しておく。学部の基礎講義として用いる際には、前半で1章から3章「整列」までをまずしっかりと教え、その後理解度に応じて4章「集合に関する操作」と5章「平衡二分探索木」に進む。そして余裕があれば6章「古典的アルゴリズム」のうちの可能な項目を教えればよい。その際には、付録に回した部分はやるとしても紹介程度で、解説している時間はないであろう。もし通年で教える場合には、演習問題などをじっくりとやら

†1 例えば「JST CREST『ビッグデータ時代に向けた革新的アルゴリズム基盤』研究代表者：加藤直樹 関西学院大学教授」など。

†2 内容的にやや高度と思われる本書の一部 (アスタリスク (*) のある項タイトルの内容) を付録として Web ページに回した。また、本書各章末の演習問題の解答も併せてこちらに掲載しているので (プログラム演習の解答は掲載していない)、コロナ社 Web ページの本書の紹介ページ

<http://www.coronasha.co.jp/np/isbn/9784339027020/>

を必要に応じて参照してほしい。なお、PDF データの入った Zip ファイルを展開する際のパスワードは以下のとおりである。

せ、付録の部分も解説することができると思う。その上で6章のすべてを教えられると思うが、さらに7章「定数時間アルゴリズム」もざっと教えられれば理想である。ただし、正則性補題はそれを理解するだけでも（一部の優秀な学生を除き）学部生には難しいので、証明は省いてアルゴリズムのみ説明することと十分である。

大学院の講義に用いる場合には、前半で1章～6章のトピックから選んで講義し、後半は7章を中心とするのがよい。付録に回した正則性補題の証明まできちんと理解させようとすると、実は7章全部で半期の講義にできるぐらいの内容はある。

本書を執筆するにあたって多くの方に助けていただいた。電気通信大学名誉教授で先進アルゴリズム研究ステーション教授でもある富田悦次先生には、本書の執筆へお誘いいただいたのをはじめとして、さまざまな機会にご助言をいただき、大いに助けていただいた。ここに深く感謝したい。また、他の方々にもたいへんご多忙であるにもかかわらず拙著の原稿に目を通していただき、間違いのご指摘や改善案の提示などいろいろご教授いただいた。たいへんありがたく思い、深謝している。それは大阪府立大学の宇野裕之准教授、京都大学学術メディアセンターの宮崎修一准教授、国立情報学研究所の吉田悠一准教授、京都大学大学院情報学研究科の玉置卓助教である。それにもかかわらずまだ不完全な部分やもしかしたらまだ間違いも残っているかもしれない。しかしそれらの責任はすべて筆者である私にある。また、コロナ社には私の遅筆でたいへんご迷惑をおかけした。ここにお詫びを申し上げる。最後になったが、分野は違えどわが尊敬する学者であり、最愛の妻である、愛知県立大学日本文化学部国語国文学科教授の伸江に心から感謝の気持ちを表したい。ありがとう。

本書が一人でも多くの方の助けとなることを祈りつつ筆を置くことにする。

2017年8月

伊藤 大雄

目 次

1 はじめに

1.1	アルゴリズムとデータ構造の重要性	1
1.2	計算モデルと計算量	4
1.2.1	問題と問題例	4
1.2.2	計算機モデル	4
1.2.3	アルゴリズムのオーダー表記	5
1.2.4	指数関数と多項式関数の比較	9
1.2.5	計算量とアルゴリズムの種類	10
1.3	NP 完全性	11
1.3.1	P と NP	11
1.3.2	NP の定義	13
1.3.3	多項式時間帰着と NP 完全	14
	演習問題	19

2 基本的データ構造

2.1	配 列	20
2.2	線形データ構造	21
2.2.1	配列を使う方法	21
2.2.2	リ ス ト	22
2.2.3	ス タ ッ ク	24
2.2.4	キュー (待ち行列)	25
2.3	木	26
2.3.1	一般的木構造	26
2.3.2	完全二分木	28

2.4	グ ラ フ	30
2.4.1	グラフの基本	30
2.4.2	次数とカット	31
2.4.3	部分グラフと路と連結性	32
2.4.4	木と森と DAG と二部グラフと完全グラフ	33
2.4.5	グラフのデータ構造	36
2.4.6	グラフの探索 — 幅優先探索と深さ優先探索	38
	演 習 問 題	40
	プログラム演習	41

3 整 列

3.1	整列とはなにか	42
3.2	バブルソート	43
3.2.1	バブルソートのアルゴリズム	43
3.2.2	バブルソートの計算時間	44
3.3	マージソート	45
3.3.1	マージソートのアルゴリズム	45
3.3.2	マージソートの計算時間	48
3.4	クイックソート	48
3.4.1	クイックソートのアルゴリズム	48
3.4.2	クイックソートの計算時間	50
3.5	バケットソート	51
3.5.1	バケットソートのアルゴリズム	51
3.5.2	バケットソートの計算時間	53
3.6	基数ソート	53
3.6.1	基数ソートのアルゴリズム	53
3.6.2	基数ソートの計算時間	54

3.7 ヒープソート	54
3.7.1 ヒープとはなにか	54
3.7.2 ヒープの構造	55
3.7.3 INSERTの方法	56
3.7.4 DELETEMINの方法	58
3.7.5 DELETEの方法	60
3.7.6 ヒープソートのアルゴリズム	61
3.7.7 ヒープの線形時間作成法	62
3.8 整列計算時間の下界値	65
演習問題	66
プログラム演習	67

4 集合に関する操作

4.1 主な操作とデータ構造	68
4.2 辞 書	69
4.2.1 ハッシュ表	69
4.2.2 外部ハッシュ法	70
4.2.3 内部ハッシュ法	71
4.3 カッコウハッシュ	72
4.3.1 カッコウハッシュとはなにか	72
4.3.2 カッコウハッシュの詳細	73
4.3.3 格納列の閉路と単純格納列	75
4.3.4 カッコウハッシュの解析*	77
4.4 ユニオン・ファインド	78
4.4.1 問題設定	78
4.4.2 配列による実現	78
4.4.3 ポインタでの実現	79
4.4.4 木による実現 — ほぼ線形時間のユニオン・ファインド	79
4.4.5 木構造上に限定された場合の線形時間ユニオン・ファインド	83
演習問題	85

プログラム演習 85

5 平衡二分探索木

5.1 平衡二分探索木の基本 86

 5.1.1 二分探索 86

 5.1.2 二分探索木の構造と MEMBER 87

 5.1.3 最大値・最小値の発見と整列 90

 5.1.4 データの挿入と削除 91

 5.1.5 回転操作 94

5.2 二色木 95

 5.2.1 二色木の基本 95

 5.2.2 二色木における挿入 96

 5.2.3 二色木における削除 98

 5.2.4 二色木の合併と分割 101

 5.2.5 別のデータの管理 104

5.3 スプレー木 107

 5.3.1 オンラインアルゴリズムと動的最適性予想 107

 5.3.2 スプレー木のアルゴリズム 109

 5.3.3 スプレー木の $O(\log n)$ 競合化* 113

5.4 タンゴ木 113

 5.4.1 タンゴ木の基本思想とインターリーブ限界 113

 5.4.2 タンゴ木の構造 117

 5.4.3 補助木のつくり替え 119

 5.4.4 タンゴ木の計算時間の解析* 122

演習問題 123

プログラム演習 123

6 古典的アルゴリズム

6.1 最小木問題	124
6.1.1 問題の定義	124
6.1.2 貪欲算法とクラスカルのアルゴリズム	125
6.1.3 クラスカルのアルゴリズムの正当性	127
6.2 最短路問題	131
6.2.1 最短路問題とはなにか	131
6.2.2 最短路の存在条件	132
6.2.3 最短路木	134
6.2.4 ダイクストラ法	137
6.2.5 フロイド・ワーシャル法	144
6.3 彩色問題	148
6.3.1 平面グラフとその性質	149
6.3.2 グラフ彩色問題	153
6.3.3 1, 2彩色問題	155
6.3.4 染色数とその上限	156
6.3.5 四色定理*	157
演習問題	157

7 定数時間アルゴリズム

7.1 定数時間アルゴリズムとはなにか	158
7.1.1 定数時間アルゴリズムの一般的枠組み	158
7.1.2 性質検査	159
7.1.3 グラフの「性質」	160
7.1.4 グラフ G と性質 P の距離	161
7.1.5 インスタンスの表現法	162
7.1.6 検査アルゴリズムと検査可能性	165

7.2 隣接行列モデル	165
7.2.1 無三角性検査	165
7.2.2 手続き TRIANGLEFREE の正当性の証明	168
7.2.3 一般化 — 無 H 性とモノトーン性	169
7.3 次数制限モデル	172
7.3.1 次数制限モデルの基本	172
7.3.2 無三角性検査と無 H 性検査	173
7.3.3 無閉路性検査	175
7.3.4 マイナー閉鎖な性質と超有限性と分割定理	179
7.3.5 分割神託	182
7.3.6 無 H マイナーなグラフの検査アルゴリズム	188
演習問題	189
プログラム演習	190

8 数学用語の解説

8.1 基本用語	191
8.2 対応・関係・関数・順序	193
8.3 基本公式	194
8.4 グラフマイナー	196
8.4.1 クラトウスキーの定理	196
8.4.2 グラフマイナー定理	198
8.5 正則性補題	200
8.5.1 正則性補題とはなにか	200
8.5.2 正則性補題の証明*	202
演習問題	202
引用・参考文献	203
索引	208

記 号 表

$\deg_G(v)$, $\deg(v)$	頂点 $v \in V[G]$ の次数	p.31
$E[G]$	グラフ G の辺集合	p.30
$E_G(U, W)$, $E(U, W)$	U と W におのおの端点をもつ辺集合	p.32
$e_G(U, W)$, $e(U, W)$	$ E_G(U, W) $, $ E(U, W) $ のこと	p.32
$E_G(U)$, $E(U)$	カット ($E_G(U, V - U)$, $E(U, V - U)$ のこと)	p.32
$\text{Ex}[X]$	確率変数 X の期待値	p.195
$\text{Pr}[*]$	事象の起きる確率	p.195
$P_T(v)$	根 (s とする) 付き出木 T の辺のみを使った $s-v$ 路	p.136
$G(W)$	頂点部分集合 W によるグラフ G の誘導部分グラフ	p.32
K_n	n 頂点完全グラフ	p.35
K_{n_1, \dots, n_k}	(n_1, \dots, n_k) 頂点完全 k 部グラフ	p.36
\ln , \lg , \log	前の二つは, 前から順に底が e , 2 の対数, 最後のものは, 底が略されている場合は, 底が 2 の対数	p.6, 7, 193
$\exp(x)$	指数関数 e^x	p.193
$O(*)$	関数のオーダー表記	p.5
$o(*)$	関数のオーダー表記	p.8
$\omega(*)$	関数のオーダー表記	p.8
$\Omega(*)$	関数のオーダー表記	p.8
$V[G]$	グラフ G の頂点集合	p.30
$\chi(G)$	グラフ G の染色数	p.156
$\Delta(G)$	グラフ G の最大次数	p.156
$\Gamma_G(v)$, $\Gamma(v)$	頂点 $v \in V[G]$ の隣接点集合	p.32
$\Gamma_G^-(v)$, $\Gamma^-(v)$	頂点 $v \in V[G]$ に入る有向辺による隣接点集合	p.32
$\Gamma_G^+(v)$, $\Gamma^+(v)$	頂点 $v \in V[G]$ から出る有向辺による隣接点集合	p.32

\times	(集合の場合) 直積, 別名デカルト積	p.193
$[\cdot], \lceil \cdot \rceil$	床関数, 天井関数。それぞれ順に, それ以下の最大, あるいはそれ以上の最小の整数を対応付ける関数	p.29, 30
\vee	論理和 OR の意味。 \cup とも書く	p.12
\rightleftharpoons	データの中身を入れ替える操作	p.43
\oplus	二つの n 頂点グラフに対する辺集合の対称差集合	p.161
2^A	A の冪集合	p.192
$\binom{n}{k}$	n 要素からなる集合から, 異なる k 個の要素を選び出す組合せの数 (二項係数)	p.192
\rightarrow	対応	p.193
\preceq	多項式時間帰着可能	p.15
\leq	順序または半順序	p.194
(A, \leq)	A は順序付き集合	p.194
$A(m, n)$	アッカーマン関数	p.80
$\alpha(m, n)$	アッカーマン関数の逆関数	p.79, 80
$e(A, B)$	AB 間の辺の本数	p.200
$d(A, B)$	(A, B) の密度	p.201

C 1 はじめに

COMPUTER SCIENCE TEXTBOOK SERIES □



1.1 アルゴリズムとデータ構造の重要性

n (> 0) 人から成るグループにおいて、各人の 100 メートル走のタイムがわかっている。その情報を基に、いろいろな基準に従って k ($< n$) 人のグループを選び出すことを考える。簡単のため、各人には 1 から n までの ID (識別番号) が割り当てられていて、 i さんのタイムは t_i だとする (表 1.1 参照)。

表 1.1 100 メートル走のタイムの表

ID	1	2	3	4	5	6	7	...	n
タイム [秒]	13.5	15.2	14.1	17.0	16.5	12.7	14.4	...	16.6

問題 1 k 人の平均タイムが最も早くなる様を選ぶ。

このとき誰でも考える当り前のやり方は、タイムの速い順から k 人選ぶ方法で、実際これで目的は果たせる。最も速い人を一人選ぶのは全員のタイムを一通り眺めればわかる。2 番目、3 番目、..., k 番目も同じなので、計算の手間は大体 kn に比例する時間になる。これを $O(kn)$ 時間と書くことにする (厳密な定義は 1.2.3 項参照)[†]。これはかなり大きな n に対しても十分速い。

では今度はつぎの条件ではどうだろうか。

[†] あるいは全員をタイムの順に並べ替えるのは $O(n \log n)$ 時間でできるので (詳細は 3 章参照), そうしておいて上から k 人選ぶ方法をとれば $O(n \log n)$ 時間でできる。 $k > \log n$ の場合はこちらのほうが有効である。

2 1. はじめに

問題 2 k 人のグループの平均タイムがグループ全体の平均タイムに最も近くなる（差の絶対値が最小になる）ように選ぶ。

グループの平均タイム \bar{t} は n 個の数値を足して n で割るだけなので、ただか $n + 1$ 回の演算で簡単に（すなわち $O(n)$ 時間で）求められる。しかし平均タイムが \bar{t} に最も近いように k 人選ぶのは簡単ではない。この場合、「 t_i が \bar{t} に近い k 人を選ぶ」というヒューリスティクスがまず思い付くが、これは最適になるとはかぎらない。

例えば、表 1.1 が見えている 8 名だけのグループ（すなわち $n = 8$ ）だったとする。このときの全体の平均タイムは 15.0 秒となる。 $k = 2$ だったとすると、平均タイムに近い二人は 2 番（15.2 秒）と 7 番（14.4 秒）だが、その平均は 14.8 秒である。しかし最適解は 1 番（13.5 秒）と 5 番（16.5 秒）の組で、平均はちょうど 15.0 秒に等しくなる。

この問題を確実に解く方法はある。 k 人からなるすべての部分集合を列挙して、その一つ一つに対して平均タイムを計算して、比較し、 \bar{t} との差が最小のものを選ぶ方法である。これならば確実に目的のものが得られるが、 n 人から k 人を選ぶ選び方は

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

通りあり、指数関数であるので、あまり大きくない n と k であっても非常に巨大になってしまい[†]、少し大きな n と k だと、この方法では、たとえ計算機を使っても解くことはできないだろう。

ではつぎの問題はどうか。

問題 3 タイムのバラツキ、すなわち分散が最小になるように k 人選ぶ。

これも問題 2 のように部分集合を総列挙すればもちろん求められるが、この場合ももっとうまい方法がありそうだ。タイムが近いもの同士を組ませればよいは

[†] 例えば $n = 40$, $k = 10$ で 3.0×10^{15} (3000 兆) 以上。

ずなので、全員をタイムの順で並べ替え、そこで連続する k 人のグループのすべてに対して確かめればよい。並べ替えで $O(n \log n)$ 時間（詳しくは 3 章 参照）、グループの数はたかだか $n - k + 1$ なので、ナイーブにやっても $O(nk + n \log n)$ 時間^{†1} で済む。これは問題 1 の解法とほとんど変わらない。

上で見たように、三つの問題に対して、それぞれいろいろな解法があった。これらが「アルゴリズム (algorithm)」である。アルゴリズムの定義は「与えられた問題を解くための、機械的操作からなる有限の手続き」などとされる。しかしそういわれてもよくわからない読者も多いであろう。具体的に問題とアルゴリズムのペアをいくつも見ていくことによって、その概念を体得していくのが一番である。

アルゴリズムにはよし悪しがある。それを測る物差しはいろいろあるが、最も標準的なのは「計算時間」である。それはすなわち、上でときどき述べていた「計算時間が $O(*)$ 」というもののことである。この関数の増加率が少ないものほど、よいとされる^{†2}。

例えば問題 3 の解法として、問題 2 のように指数関数的に時間のかかるアルゴリズムをつくってしまうと、ごく小さい n と k の場合は別として、ほとんど役に立たない^{†3}。もし君がシステムエンジニアでそんなプログラムをつくったとしたら、会社の業務は進まず、君は早晚クビになるだろう。しかし少し工夫すれば、問題 1 を解くのとあまり変わらない計算時間でできることになり、会社も繁栄し、君のボーナスも増えるかもしれない。

これがアルゴリズムの重要性である。

アルゴリズムをプログラム上で実現するにはデータ構造が重要である。データ構造とはデータを表現する方法であり、それが適切かどうかでアルゴリズムの性能が大きく変わる。データ構造の重要性を説明するには、もう少し突っ込

†1 この意味は、 nk と $n \log n$ のうち大きいほうに比例する時間がかかる、ということ。詳しくは 1.2.3 項 参照。

†2 データ量が増えても計算時間が爆発しにくいからである。

†3 一時的に使うだけで、しかもごく小さい n と k に対してだけしか使用しないとわかっている場合は、サッと書けるのでこういうナイーブなアルゴリズムもあり得る。

んだ知識が必要となるので、ここでは述べないが、例えば上のアルゴリズムの計算時間が $O(kn)$ 時間などと述べたが、データ構造が不適切だとその性能は出ない場合がある。

1.2 計算モデルと計算量

アルゴリズムと計算時間を理論的に扱うためには、厳密なモデル化が必要となる。

1.2.1 問題と問題例

上でも「問題 1」などのように「問題」を扱ったが、アルゴリズムと計算量の理論研究の分野で扱う問題 (problem) とは、通常無限個の問題例 (problem instance) から成っている。例えばつぎの問題を考える。

分割問題

入力 n 個の整数 a_1, \dots, a_n

要請 a_1, \dots, a_n を総和が等しい二つの部分に分けられる (すなわち $\sum_{i \in B} a_i = \sum_{i \notin B} a_i$ である $B \subseteq \{1, \dots, n\}$ が存在する) か否かを判定せよ。

この問題において、 n や a_1, \dots, a_n を具体的な数値として与えたものを問題例と呼ぶ。問題はこうした問題例が無限個集まったものとして定義される (演習問題【1】 (p.19) 参照)。

1.2.2 計算機モデル

計算機モデルは基本的に RAM (random access machine, ランダムアクセスマシン) と呼ばれる、「ランダムアクセス記憶をもつ、逐次型計算機」を想定する。理論的に扱いやすくするために、以下の仮定を用いる。

- メモリの 1 セルとレジスタには、任意に大きい数を格納できる。

- 各命令（四則演算・比較・メモリ 1 セル分のアクセスなど）は単位時間で実行できる。
- メモリのセル数は無限

これはすなわち実在のコンピュータの理論抽象化である。アルゴリズムの計算時間は、この計算機における計算時間、すなわち命令が実行された数（計算ステップ数）で評価される。

1.2.3 アルゴリズムのオーダー表記

(1) ビッグオー表記の定義 アルゴリズムの計算時間（計算ステップ数）は入力の大きさとの関連下で評価されなければならない。すなわち、問題例の入力に必要なデータ量 n の関数 $T(n)$ として、アルゴリズムの計算時間を表す。

ただし、その計算時間をあまり細かく算出しても意味はない。プログラムの組み方によって、定数ステップ程度の違いは出てくるし、それらは当然本質的ではない。そのため、アルゴリズムの計算時間を評価する際には前述の $O(n \log n)$ などという表記を用いるのが普通である。このような表記を「オーダー表記」あるいは「ビッグオー表記」などという。ここで、この表記について正確に定義を与えておく。なお、多くのアルゴリズム開発者にとっては、ここで記する正確な定義は覚えておく必要はなく、その後で述べる「大まかなやり方」を知っておけば十分である。

● **定義 1.1** 二つの関数 $f(n)$, $g(n)$ について $f(n) = O(g(n))$ であるとは、ある整数 $m > 0$ と実数 $c > 0$ が存在して、任意の $n \geq m$ に対して $f(n) \leq cg(n)$ となることである。

この定義に従って $f(n) = 3n^3 + 10n^2 + 5n \lg n - 30n + 100 = O(n^3)$ であることを確かめてみよう。例えば、100 以上の n に対して明らかに $f(n) \leq 100n^3$ であるので、 $m = 100$, $c = 100$ とすれば、定義 1.1 の条件を満たすので、

索引

【あ】

アーク	31
頭	31
後入れ先出し	24
跡取り	114
アルゴリズム	3

【い】

位数	181
位相的マイナー	197
一始点問題	132
一対一対応	193
一対問題	131
遺伝的	172
入木	35
入次数	32
インターリーブ限界	116

【え】

枝	30
---	----

【お】

尾	31
オイラーの多面体公式	151
オーダー表記	5
オフラインアルゴリズム	108
重み	55
親	27
オンラインアルゴリズム	108

【か】

開区間	191
回転	94
外部ハッシュ法	70

格納列	74
片側誤り	165
カッコウハッシュ法	70, 72
カット	32
合併	101
関係	193
関数	193
完全 k 部グラフ	36
完全グラフ	35

【き】

木	34
基数ソート	53
逆対応	193
キュー	25
競合比	108
強連結	33
極大森	34
禁止マイナー	199
均等分割	201

【く】

クイックソート	48
クラスカルのアルゴリズム	125
クラトウスキーの定理	196, 197
グラフ彩色問題	153
グラフマイナー定理	199

【け】

計算機モデル	4
計算複雑さ	10
計算量	10
決定問題	14

検査アルゴリズム	165
検査可能	165

【こ】

子	27
---	----

【さ】

最悪計算量	10
最小木問題	124
彩色	154
彩色関数	154
彩色問題	154
最短路	131
最短路木	135
最短路部分木	136
最短路問題	131
最適性の原理	136
細分	197
先入れ先出し	25

【し】

時間計算量	10
自己ループ	31
辞書	69
次数	31
次数制限モデル	164
子孫	27
質問計算量	163
写像	193
充足可能性問題	12
縮約	198
順序	194
順序付き集合	194
順列	192
初等的	33

【す】

推移律 194
 スタック 24
 スプレー木 109
 スマートオーダー 9

【せ】

性 質 161
 正則性補題 166, 200, 201
 整 列 42
 整列問題 42
 接続する 30
 節 点 30
 セパレーション 180
 全域木 34
 全域部分グラフ 32
 全域森 34
 線形順序 194
 全 射 193
 全順序 194
 染色数 156
 全単射 193
 全対問題 132

【そ】

疎 163
 双方向リスト 23
 疎グラフモデル 164
 祖 先 27

【た】

対 応 193
 ダイクストラ法 137
 高 さ 27
 多項式時間帰着可能 15
 多重グラフ 31
 タンゴ木 114
 探 索 38
 単 射 193
 単 純 33, 75
 単純グラフ 31
 単純 3 分割問題 19

単純ナップザック問題 16
 端 点 30, 33

【ち】

チェルノフ限界 195
 嫡 流 117
 頂 点 30
 直 積 193

【て】

定数時間アルゴリズム 158
 デカルト積 193
 出 木 34
 出次数 32

【と】

同 型 160
 動的最適性予想 109
 特性関数 68, 191
 貪欲算法 125

【な】

内部ハッシュ法 70, 71
 長 さ 33, 131

【に】

二項関係 193
 二項係数 192
 二色木 95
 二部グラフ 35
 二分探索 86
 二分探索木 87

【ね】

根 34, 35
 根付き木 35
 ネットワーク 124

【は】

葉 27, 34, 35
 配 列 20
 バケットソート 51
 ハッシュ関数 69

幅優先探索 39
 バブルソート 43
 反射律 194
 半順序 194
 反対称律 194
 判定問題 159

【ひ】

ヒープ 54
 ヒープ条件 56
 ヒープソート 54
 比較可能性 194
 ビッグオー表記 5

【ふ】

部 35
 フィボナッチヒープ 144
 深 さ 27
 深さ優先探索 39
 部分グラフ 32
 プリムのアルゴリズム 125
 ブルックスの定理 157
 フロイド・ワーシャル法 144
 分 割 101
 分割神託 183
 分割定理 181
 分割問題 4

【へ】

平均計算量 10
 閉区間 191
 平衡二分探索木 95
 平面グラフ 149
 平面的グラフ 150
 並列辺 31
 閉 路 33, 75
 ヘーフディングの不等式 195
 べき集合 192
 冪集合 192
 辺 30
 辺オラクル 164

NP 困難	18	ρ 超有限	180	v_0-v_p 路	33
n 頂点グラフ	160			$v-r$ 路	27
		[S]		v を根とする部分木	28
[P]		SAT	12		
P	11	$s-t$ 最短路	131	[Z]	
		[V]		zig-zag 型	110
[R]				zig-zig 型	110
RAM	4	v_0-v_p 間の路	33	zig 型	109

—— 著者略歴 ——

1985年 京都大学工学部数理工学科卒業
1987年 京都大学大学院修士課程修了（数理工学専攻）
1987年 日本電信電話株式会社基礎研究所
1990年 日本電信電話株式会社通信網総合研究所
1995年 博士（工学）（京都大学）
1996年 豊橋技術科学大学講師
2001年 京都大学助教授
2007年 京都大学准教授
2012年 電気通信大学教授
現在に至る

データ構造とアルゴリズム

Data Structures and Algorithms

© Hiroo Itoh 2017

2017年9月28日 初版第1刷発行

検印省略

著者 伊藤 大雄
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 978-4-339-02702-0 C3355 Printed in Japan

(金)



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 03-3513-6969, FAX 03-3513-6979, 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。