P区間表：区間に関するプログラム作成問題のためのデータ構造

二村良彦[†1] 白井千恵子[†2]
劉咏梅[†3] 二村夏彦[†4]

数列の区間および数平面上の矩形の中で，与えられた性質を有する最長あるいは最大なものを求める実用的なプログラム作成問題は多い．これらの問題はプログラミングの教科書等においても頻繁に取りあげられるが，多くの学生が容易に理解可能な解法が知られていなかった．本稿ではこれらの問題のサブクラスを効率良く一様に解決する目的でP区間表と呼ばれるデータ構造を提案する．そしてそれが$O(n)$時間およびスペースで作成できるための条件と作成法を示す．さらに，P区間表がプログラム作成問題の理解しやすい解法を与えることを，大学のプログラミング演習を通じて検証したので，その方法と結果について報告する．最後にP区間表の広範な適用能力を示すために，新問題とP区間表に基づくその最適解法を与える．

P-segment Tables: Data Structures for Programming Problems Concerning Segments

YOSHIHIKO FUTAMURA,[†1] CHIEKO SHIRAI,[†2] YONGMEI LIU[†3] and NATSUHIKO FUTAMURA[†4]

There are many practical programming problems finding longest segments or largest rectangles in a given sequence or plane. Several text books on programming proposed solutions for the problems. However, only a few solutions of them are easy enough to be understood by ordinary computer science students. This paper proposes new data structure *P*-segment tables for solving programming problems concerning segments or rectangles. We give a condition and an algorithm for making a *P*-segment table in $O(n)$ time and space. After that, several problems are solved by using the proposed methods. We have conducted a survey through class exercises, examinations and questionnaires in order to prove that the programming method based upon *P*-segment tables is understandable, applicable to various problems and favorable for students. The survey methods and results are described. Finally, an optimal solution by *P*-segment table for an open problem is given.

1. はじめに

数列の区間および数平面上の矩形の中で，与えられた性質を持つ最長あるいは最大なものを求めるプログラム作成問題は，図形処理や半導体設計等，多くの実用的問題の中に現れる．そのうえこれらは問題点が明確でありかつその解法の複雑さが必要な計算量に応じて変化する．したがって，プログラミングに関する重要な概念の1つである計算量を習得するのに適しており，プログラミングの教科書等においても演習問題として取り上げられている[1]．しかし，そこで与えられている解法は特殊で汎用性がなかったり[11]，あるいは，形式的で具体性に欠けている[1]．そのため，我々の教育経験によれば，多くの学生は示された解法を理解すること，およびその他の類似の問題への適応に困難を感じている．また，スタックなどの既存のデータ構造を用いた理解しやすい統一的解法は，筆者らの調査の範囲では見い出すことができなかった[7),8)]．

我々は数列の区間や数平面上の矩形に関するプログラム作成問題のためのデータ構造としてP区間表を提案し，P区間表を用いて，これらの問題のサブクラスの効率の良い統一的解決法を得た．P区間表とは，

†1 早稲田大学理工学部
School of Science and Engineering, Waseda University
†2 早稲田大学大学院理工学研究科
Graduate School of Science and Engineering, Waseda University
†3 武漢大学
Wuhan University
†4 School of Computer and Information Science, Syracuse University

数列の各点に対しそれを端点とし，かつ性質 P を有する最短（または最長）の区間を表す表である．さらに，P 区間表に関する教育上の有効性を調査するため，1994 年度および 1995 年度に筆者らがプログラミング教育を担当した大学学部 2 年生各約 100 名に P 区間表とその利用法を教育した．そして演習，試験およびアンケートを通してその教育効果を調査した．その結果，P 区間表を用いる技法は数列の区間長や数平面上の矩形面積の最大化等のプログラム作成問題に関する限り，学生が理解し，受け入れかつ応用しやすい方法であることが検証できた．

本稿では，まず P 区間表とそれが $O(n)$ 時間およびスペースで作成できるための条件と作成法について述べ，その適用例を示す．次に P 区間表の有効性の検証方法と結果について報告する．最後に P 区間表を用いた新問題（最長平区間問題[6]）の最適解法を与え，その適応能力の広さについて議論する．なお，本稿では特に断らない限り，問題となっている数列の長さは n と考える．また，アルゴリズムの記述には筆者らが実際に本稿で述べる教育において使用したプログラム図式 ISO8631 PAD[3]を用いる．PAD の説明は付録 A.1 で行う．

2. P 区間表とその作成法

プログラミングにおいて，重複計算を避けるために表を利用する技術は一般的にダイナミックプログラミングと呼ばれている[5]．ここでもその技術を応用し，区間に関する表を利用する．以下では P 区間表の定義とそれが $O(n)$ 時間およびスペースで作成可能な場合の条件と作成法を示す．

2.1 P 区間表

P 区間表 t とは，与えられた数列 x 上の各点 i $(1 \le i \le n)$ を左端（または右端）とし，与えられた関係 P を満足する最短区間の右端 j $(i \le j \le n)$（もしくは左端 j $(1 \le j \le i)$）を示す表である☆．ただし，P を満足する右端（または左端）が存在しない場合には $t[i]$ には $n+1$ または nil を入れる．右端を対応させた表を右 P 区間表，左端を対応させた表を左 P 区間表と呼ぶ．すなわち，右 P 区間表は**表 1** のような構造を有する．以下では特に断らない限り，右 P 区間表を P 区間表または単に区間表と呼ぶ．また区間 $x[i]\ldots x[j]$ を $[i,j]$ と表す．たとえば数列 6,3,5,8,6,3 に対し，関係 $P(i,j)$ を「右端 $x[j]$ は左端 $x[i]$ より小さい」とすると，対応する（最短右 P）区間表は**表 2** のようになる．また，数列 11010010 に対して，$P(i,j)$ を「$[i,j]$ は均衡区間である」とすると，対応する（最短右 P）区間表は**表 3** のようになる．ただし，均衡区間とは，0 と 1 のみからなる数列において 0 と 1 の個数が等しい区間である．

☆ $P(i,j)$ を満足する最大の右端 j を対応させる最長区間表に関する議論も同様に行えるが，紙面の都合で割愛する．

表 1 右 P 区間表
Table 1 Right P-segment table.

座標	1	...	i	...	n
値	$x[1]$	...	$x[i]$	...	$x[n]$
右端	$t[1]$	...	$t[i]$	...	$t[n]$

ただし，$t[i] = \min\{j \mid P(i,j) \wedge i \le j \le n\}$

表 2 右区間表の例—数列 6,3,5,8,6,3 に対する右隣の小さい数
Table 2 Example of a right segment table.

座標	1	2	3	4	5	6
値	6	3	5	8	6	3
右端	2	nil	6	5	6	nil

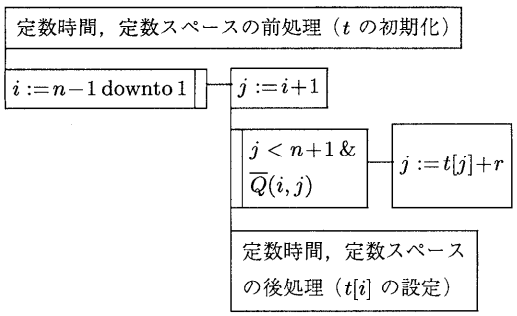
ただし，表中の nil は $n+1$ を表す．

表 3 右区間表の例—数列 11010010 に対する最短均衡区間
Table 3 Example of a shortest balanced segment table.

座標	1	2	3	4	5	6	7	8
値	1	1	0	1	0	0	1	0
右端	6	3	4	5	nil	7	8	nil

2.2 P 区間表の作成法

区間表の一番素朴な作成法は次のとおりである．すなわち，各点 i から右に向かって j を 1 ずつ増やしながら $P(i,j)$ を調べ，最初に P が成立する j を $t[i]$ に入れればよい．関係 P の計算量を $O(k)$ と仮定すると，この場合に区間表を作成するための計算量は明らかに $O(kn^2)$ である．この計算量で区間表が作成できても，計算量が多すぎてありがたみは少ない．しかし，我々は区間表を $O(n)$ 時間およびスペースで作成できる次のような場合を発見した．それは，定数時間およびスペースで計算できるある関係 $Q(i,j)$ が存在し，ある整数定数 r（ただし，$r \ge 0$ かつ $\overline{Q}(i,j)$ なら $t[t[j]+r] > t[j]+r$）に対して，所望の区間表が**図 1** のアルゴリズム（ISO PAD[3]による記述）により作成できる場合である．たとえば，表 2 に対する例では，$r=0$ そして $Q(i,j) \equiv (x[i] > x[j])$ とすればよい．ただし，$t[n] := n+1$ を前処理の部分で，そして $t[i] := j$ を後処理の部分で実行する．また表 3 に対しては，$r=1$ そして $Q(i,j) \equiv (x[i] \neq x[j])$ とすればよい．ただし，$t[n] := n+1$ を前処理の部分で，そして $t[i] := j$ を後処理の部分で実行する．(r と Q のこの設定で表 3 のような最短均衡区間表が作成できることは自明でないので，その証明を付録 A.2



ただし，$r \geq 0$ かつ $\overline{Q}(i,j)$ なら $t[t[j]+r] > t[j]+r$

図1 P 区間表作成アルゴリズム（ISO PAD[3]による記述）
Fig. 1 Algorithm for making a right P-segment table.

に示した.）このアルゴリズムが $O(n)$ 時間およびスペースであることの証明は付録 A.3[2),4)]に示した．ただし，後処理では $t[i]$ の設定以外の操作を行わないものとする（左 P 区間表も同様に作成できる）.

長さ n の数列の区間に関するプログラム作成問題では，数列全体を少なくとも1回は調べるので，ほとんどすべての場合に作成されるプログラムの計算量は $O(n)$ 以上である.

したがって区間表が $O(n)$ 時間およびスペースで作成できれば，それは全体の計算量を増加させる要因にはならない．次章で示すとおり，区間表の利用により多くのプログラム作成問題を統一的方法でしかも能率良く解決することができる．図1に示した我々の方法は，いかなる P に対しても有効であるとは限らないという意味で完全ではない．我々はその方法が適用可能な関係 P のクラスに関する調査を行っているが，まだ明快な結論は得られていない．理論的な解明は今後の課題である．また関係 Q および定数 r はプログラマ自身が発見しなければならない．しかし，区間表を作成するうえでそのプログラム構造が図1により明快になり，しかも関係 P から Q を導けば $O(n)$ のアルゴリズムが得られるというガイドラインが，プログラマにとって明確になる．それゆえ，5章で示すように，我々のアプローチは現在の段階においても実際的に有意義であることを，教育の現場で検証できた.

3. P 区間表の適用例

この章では区間表の適用例として，数列の区間や数平面上の矩形に関するプログラム作成問題とその解法を示す.

3.1 右隣の小さい数の表作成問題

右隣の小さい数の表とは，数列 x の各点 i に対し，i よりも右にある $x[i]$ よりも小さい要素の中で最も i に近い要素 j を対応させた表である．問題はこの表を $O(n)$ 時間およびスペースで作成することである.

すでに表2の例で述べたように，この問題において $P(i,j)$ は「右端 $x[j]$ は左端 $x[i]$ より小さい」である．したがって，前述のとおり図1において $r=0$ そして $Q(i,j) \equiv (x[i] > x[j])$ とすればよい．この問題をここでもう1度取り上げた理由は，(1) 右隣の小さい数の表は多くの数列の区間問題に適用可能，かつ (2) 適度の難しさを持った格好のプログラム作成問題だからである．実用的にもこれは，対称ヒープの作成[2)]，ソーティング[2)]，多角形の三角形分割等に現れる重要な問題である.

左隣の小さい数の表も左区間表として上と同様に作成できる．この2つの表があれば，$x[i]$ が最小である最長の区間を定数時間で求めることができる．以下ではこの2つを合わせて最小区間表と呼ぶ.

3.2 最大区間面積問題[11)]

区間とは数列の連続する部分列をさす．区間面積とは区間の最小値と区間の長さの積である．問題は，与えられた数列に含まれるすべての区間の中で区間面積が最大となるものを $O(n)$ で求めることである．たとえば，数列 5,1,3,5,4,2 に対しては下線が引かれた区間が区間面積が最大となる．最小値が3で区間長が3なので区間面積は9である．この問題は，実用的には3.6節の最大空部分行列問題を解くために利用される.

すべての区間には最小値があるので，最大面積を持つ区間も当然最小値を持つ．したがって，この問題は数列の各点 i に対して，その要素 $x[i]$ が最小である最長の区間 $[L,R]$ を求めることによって解決できる．最長の区間 $[L,R]$ は3.1節の右区間表 t を利用すれば $R=t[i]-1$，同様に左区間表 t' を作成すれば $L=t'[i]+1$ となる．よって最大区間面積はそのような区間の中で面積最大なものを選べばよい.

ここで注意しなければならない点がある．それは「最長な区間の長さだけを求めればよい」という問題に対して区間表を作ることは，ある意味で「やりすぎ」という場合があることである．たとえば，表の代わりにスタックを使えばスペース計算量を「つねに $O(n)$」から「最悪 $O(n)$」に変えることができる．また，時間計算量の定数を減少させることも可能である．しかし，プログラミング教育においてはまず分かりやすい統一的方法で能率の良い解を得，その後で必要に応じてそのプログラムを最適化するというアプローチをとらせることが重要である．これにより段階的詳細化の技術を習得させることができる．以下の問題で，最長あるいは最大等，1つの量を求めればよい場合にも上

と同じ議論が成立する．

3.3 最大正方区間面積問題

正方区間とは，区間の最小値とその長さが等しい区間である．問題は，与えられた数列に含まれるすべての正方区間の中で面積最大なものを $O(n)$ で求めることである．

この問題は前述の最大区間面積問題と同様に解決できる．すなわち，数列の各点 i に対して，その要素 $x[i]$ が最小である最長の区間 $[L,R]$ を求める．そのような区間の中で区間の長さが区間の最小値以上のものは，正方区間を含む．よって最大正方区間面積は，区間の長さが区間の最小値以上の区間の中で正方区間面積最大なものを選べばよい．この問題は筆者らの知る限りでは，米田信夫氏が3.2節の最大区間面積問題を見た後で作成したものである[12]．したがって実用的応用を考えて作成された問題ではない．これは前節の問題の応用問題である．米田氏による解法は知られていないが，P区間表を用いた解法ではこの問題も，前述の最大区間面積問題もほとんど同じものと見なすことができる．また低地問題[10]（その最小値が区間の幅以下の区間を低地と呼ぶ．与えられた数列に含まれる最大面積の低地を求める問題）もまったく同様に解決できる．

3.4 最長素均衡区間問題

素均衡区間は均衡区間であって，かつ左右に2分割しても空でない2つの均衡区間を得ることができないものである．均衡区間は先に述べたとおり，0と1のみからなる数列において0と1の個数が等しい区間である．問題は，与えられた0と1のみからなる数列において，長さが最大の素均衡区間を $O(n)$ かつスペースで求めることである．たとえば，数列11010010の素均衡区間最大長は，6（下線が引かれた区間の長さ）である．

この問題は，表3の最短均衡区間表を作成することによって解決できる．すなわち，$x[i]$ から始まる素均衡区間は $x[i]$ から始まる最短均衡区間と同じである．前述のとおり素均衡区間表作成において，与えられた関係 $P(i,j)$ は「$[i,j]$ は均衡区間」となる．この関係 P に対し，図1の P区間表作成のアルゴリズムにおいて，$r=1$ そして $Q(i,j) \equiv (x[i] \neq x[j])$ とすれば素均衡区間表は $O(n)$ で作成できる．表には，各点から始まる最短の均衡区間すなわち素均衡区間が示されているので，求める最長素均衡区間はそのような区間の中で最長のものを選べばよい．この問題は次節の最長均衡区間問題を解決するための部分問題として筆者らが考え出したものである．

3.5 最長均衡区間問題[1)]

問題は，与えられた0と1のみからなる数列に含まれる均衡区間の中で，最長のものを $O(n)$ で求めることである．たとえば，数列11010010の均衡区間最大長は，8（下線が引かれた区間の長さ）である．

この問題に対しては前述の素均衡区間表を利用し，次のようにして最長均衡区間表を作成すればよい．最長均衡区間表 t' は各点 i に対して，$x[i]$ を左端とする素均衡区間 $[i,j]$ に $x[j+1]$ を左端とする最長均衡区間 $[j+1,t'[j+1]]$ をつなげば $O(n)$ で作成できる．$x[i]$ から始まり，$x[t'[j+1]]$ を越えて右に伸びる均衡区間があると，それは $[j+1,t'[j+1]]$ の最長性に反する．したがって，区間 $[i,t'[j+1]]$ が $x[i]$ から始まる最長の均衡区間である．よって最長均衡区間はそのような区間の中で最長のものを選べばよい．ちなみに一般の数列に対する均衡区間の定義は「要素の総和が0の区間」である．この場合，最長均衡区間問題の解としては $O(n\log n)$ のものが知られている[9)]．この問題は，数学の有限群論において単位元に還元可能な最長の式を求める問題に対応している．

3.6 最大空部分行列問題[1)]

0と1のみからなる行列が与えられたとき，その空部分行列とは要素がすべて0である部分行列である．問題は，与えられた行列に含まれる空部分行列の中で，面積が最大なものを $O(N)$ 時間かつスペースで求めることである．ただし N は行列に含まれる要素数である．たとえば**表4**の行列に対する最大空部分行列の面積は5である（表4の行列で下線を引かれた 5×1 の部分）．実用的には，欠陥のない基板の切り出し等，この問題の応用範囲は広い．これは下記に定義する累積表を利用し，問題を最大区間面積問題に還元することにより解決できる．

定義1 M を0と1のみからなる行列とする．このとき $M(i,j)$ が0ならば，そこから上方に向かう j 列上の0の連の長さを $A(i,j)$ とする．また $M(i,j)$ が1ならば $A(i,j)=0$ とする．このようにして得られる行列 A を M の累積表と呼ぶ（表4参照）．

行列の要素数を N とするとその累積表は $O(N)$ 時

表4 行列 M（左）とその累積表 A（右）
Table 4 Matrix M and its accumulation table.

m

0	1	0	0	1	
0	0	1	1	0	
0	1	0	1	1	
0	1	0	0	1	
0	1	0	0	0	

n

1	0	1	1	0
2	1	0	0	1
3	0	1	0	0
4	0	2	1	0
5	0	3	2	1

2	17	17	7	14
17	10	15	12	1
18	4	11	10	5
6	11	18	14	18
5	7	19	13	4

図2 最大矩形体積の例
Fig. 2 Example of the largest rectangle volume.

間で作成できることは明らかである．最大空部分行列問題は与えられた行列の累積表を作成し，累積表の各行に対して最大区間面積を求め，その中から面積が最大のものを選べばよい．したがって，最大空部分行列問題も $O(N)$ 時間およびスペースで解決できる．

3.7 最大矩形体積問題

ここでは前節の問題の3次元への拡張を考える．実用的には，この問題は3次元のパッキング問題等広範な応用が考えられる．矩形とは与えられた行列の部分行列である．矩形体積とは矩形の最小値と矩形面積の積である．問題は与えられた行列に含まれるすべての矩形の中で矩形体積が最大となるものを求めることである．たとえば，**図2** の行列に対する最大矩形体積は，最小値が10で，矩形面積が8の部分で，体積は80となる（図2の下線が引かれた要素からなる 4×2 の矩形）．

この問題は筆者らが作成した問題であるが，現時点では最適解は不明である．ここではともに $O(N)$ スペースであるが（1）最悪 $O(N^{1.5})$，（2）平均 $O(N \log N)$ かつ最悪 $O(N^2)$，の2解法を示す．

3.7.1 解 法 1

与えられた行列を M とする（$N = n \times m$ かつ $n \leq m$ とする）．すべての矩形はある縦幅を持つので，最大体積の矩形も当然縦幅を持つ．したがって，まず $1 \leq x \leq y \leq n$ なる任意の x, y の組合せ（$O(n^2)$ 個）に対して，縦がちょうど x と y の間にある矩形の中から体積最大なものを選ぶ．縦幅が一定なので，長さ m の数列 $\min\{M(i,1) \mid x \leq i \leq y\}, \ldots, \min\{M(i,m) \mid x \leq i \leq y\}$ に対する最大区間面積問題を解き，その解に縦幅 $y-x+1$ を乗じたものが求める最大体積である．これを x と y のすべての組合せについて行い，その中から最大値を求めればよい．$\min\{M(i,j) \mid x \leq i \leq y\} = \min\{\min\{M(i,j) \mid x \leq i \leq y-1\}, M(y,j)\}$ であるので y に対する数列は，同じ x に対する $y-1$ の数列から $O(m)$ で求めることができる．したがって，必要なすべての数列の個数は $O(n^2)$ であるので，それは $O(mn^2)$ で求めることができる．また，各列に対する最大区間面積問題は $O(m)$ で解けるので，この手順全体の最悪計算量は $O(mn^2)$，すなわち $O(N^{1.5})$ である．

3.7.2 解 法 2

M の各点 $M(i,j)$ に対し，その値を最小値とするすべての矩形は，その点から上下左右各方向へ探して最初に出会うその点より小さい4点に囲まれる矩形に含まれる．その矩形の面積は下記の定理1より平均 $O(\log n \times \log m)$ である．この矩形を各点に対して定数時間で求めるために，前もって各行および各列について最小区間表（3.1節参照）を作っておく．さらに M の要素を降順にソートしておき，値の大きな点から M と同サイズの別の行列 M' に入れてゆく（M' の初期値はすべての要素の値が0）．すると各点 $M(i,j)$ が処理されるときには M' 上ではその点が0点を除いて最小である．ここで0点を1そしてその他の点を0と考えなおせば，問題は $M(i,j)$ を含む空部分行列を，平均サイズ $O(\log n \times \log m)$ の範囲で探せばよいことになる．$M(i,j)$ が新たに M' に入った時点で累積表における $M(i,j)$ とその上下の部分に対応するところだけ変更すれば，累積表は正しく維持できる．したがって定理1よりこの累積表の書き換え作業は $O(\log n)$ である．下記の定理2より，累積表ができていれば指定された点を含む最大空部分行列問題は平均 $O(u)$（この場合 $u = \log m$）で解ける．したがって $O(N \log m)$ で各点に対する最大矩形を求めることができる．ただしソートに $O(N \log N)$ かかるため，その時間が上限となる．

定理1 一様乱数列が与えられたとき，最大区間面積問題に現れた区間 $[L, R]$ の平均長は $O(\log n)$ である（紙面の関係で証明は省略するが，興味のある読者は文献4)を参照されたい．下記定理2についても同様）．

定理2 x 軸に関して単調な多角形 G および多角形内の x 軸上の点 A が与えられたとき，A を含みかつ G に含まれる最大正置矩形は平均 $O(u)$，最悪 $O(u^2)$ 時間および $O(u)$ スペースで求めることができる（ただし u は G に含まれる x 軸の長さ）（証明は文献4)参照）．

以上で7つの問題を P 区間表を用いて統一的方法で解決した．我々はスタックなどの既存のデータ構造を用いて同様の試みを行ったが，失敗した．その主要因は，スタックは強力なためその利用法は柔軟性に富み過ぎ，同じ問題を解く場合でも解決者の個性が強く表れる．すなわち「スタックを用いよ」という指示は，解法のガイドラインとはなり得ない．これはちょうど，プログラム図式におけるフローチャートが柔軟性に富み過ぎるため，構造化プログラミングの教育に不適で

表5 関係 P から Q を求めることは容易か
Table 5 Getting Q from P is easy.

問題＼難易度	容易	困難だが実行可能	実行不能
右隣の小さい数の表作成問題	56人（58%）	40人（42%）	0人（0%）
最大区間面積問題	70人（73%）	26人（27%）	0人（0%）
最長素均衡区間問題	68人（71%）	27人（28%）	1人（1%）

あることと類似している[7),8)]．

4. P区間表の有効性の検証方法

アルゴリズム教育上のP区間表の有効性を，その理解のしやすさ（理解度），応用のしやすさ（応用度），受け入れやすさ（好感度）の3つの観点から調査した．調査対象は，1994年度および1995年度に筆者らがプログラミング教育を担当した大学学部2年生各約100名である．学生にはP区間表に関する予備知識をいっさい与えず，演習問題を表7の順序で出題した．第1回目のレポート提出後に，P区間表の作成法とその計算量を講義した．その時点では，一部の学生は「右隣の小さい数の表作成問題」を自力で解決することにより，より一般的なP区間表を理解するための準備ができていた．区間表作成の計算量の厳密な証明を自力でできた学生は皆無であったが，一部の学生は直感的に計算量が$O(n)$であることを理解していた．学生が提出したレポートおよび期末試験に基づき，学生が区間表を実際にどれだけ理解できたかを調べた．また95年度の学生にはP区間表の有効性に関するアンケート調査（主観調査）を実施した．なお，演習問題の出題に際し，下記の条件を課した：

- 回答期限は2週間とした．
- できる限り効率の良い解法を得るように要請した．そのため採点基準を計算量に応じて設定した．最適解なら評価はA，最適解ではないならば，計算量に応じて評価をB，C，間違いはDと明示して出題した．たとえば最適解が$O(n)$ならば，$O(n)$がA，$O(n^2)$がB，$O(n^3)$がC，間違いはDという具合である．
- 計算量に関しては理論的な証明をするように要請した．

5. 調査結果

ここでは上記の調査結果について報告する．

5.1 P区間表の理解度

P区間表を$O(n)$で作成するためには，問題の中から関係Pを抽出し，かつそのPから図1の関係Qを求める必要がある．学生がこの作業をどれほど困難と感じたかを，前章で述べたプログラム作成問題のうち3つについて回答を求めた．回答は学生の主観に基づくものである．その結果は**表5**のとおりである．PからQを求めること，すなわち，P区間表の作成法はほとんどすべての学生が理解可能と判断したといえる．

表6 P区間表$O(n)$作成の証明に関する理解度
Table 6 Proof of $O(n)$ algorithm for making P-segment table is understandable.

＼理解度	理解できた	理解できない
アンケート	56人（54%）	48人（46%）
期末試験	92人（90%）	10人（10%）

プログラミングの講義においては計算量の概念を学生に教える必要がある．教えられたとおりにプログラムが作成できるだけでは不十分であり，そのプログラムの計算量も評価できるようにしなければならない．そこで，図1のアルゴリズムが$O(n)$時間およびスペースであることの証明（付録A.3）の理解度を調べ，**表6**に示した．調査はアンケート（主観調査）および期末試験（アンケート調査の1カ月後）に基づく．期末試験では，アルゴリズムの計算量の証明ができたことをもって理解できたと見なした．アンケート調査と期末試験の結果には理解度に関する差違が見られる．これは学生が試験に備えて証明法を暗記したともとれる．いずれにせよ大半の学生が理解できたと考えられる．

5.2 P区間表の応用度

学生がP区間表を理解し，他の問題解決の際にうまく応用できるか，すなわち応用度を演習問題の正答率から判断した．学生のレポートを採点し，最適解で答えた学生数とその割合を**表7**に示した．ただし表7の調査において，学生の解法がP区間表に基づかないものについては，アルゴリズムの最適性（この場合は$O(n)$）に関する判定が困難なものが多くあった（一般にアルゴリズムの最適性の証明は容易ではない）．したがって，表7に示された最適解で答えた学生数の欄で約半数と示されているのは，正確な数値が把握できなかった場合である．

第1回演習課題はP区間表に関する予備知識をまったく与えずに出題した．しかし，第1問の「右隣の小さい数の表作成問題」が，最大区間面積および最大正

表7 レポート採点結果
Table 7 Evaluation of submitted reports.

	問題 \ 最適解で答えた学生数	94年度	95年度
第1回	右隣の小さい数の表作成問題 ($O(n)$)	約半数	33人 (34%)
	最大区間面積問題 ($O(n)$)	約半数	27人 (28%)
	最大正方区間面積問題 ($O(n)$)	約半数	27人 (28%)
第2回	最長素均衡区間問題 ($O(n)$)	62* 人 (55%)	62** 人 (58%)
	最長均衡区間問題 ($O(n)$)	約半数	53*** 人 (50%)
第3回	最大空部分行列問題 ($O(N)$)	88人 (86%)	76人 (78%)
	最大矩形体積問題 ($O(N^{1.5})$)	22人 (22%)	44人 (45%)

*, **, *** はそれぞれ28人, 17人, 16人がP区間表を使わない解法

方区間面積の解法のヒントを与えるため,「右隣の小さい数の表作成問題」を最適解 ($O(n)$ の解法) で作成することができた学生のほとんどすべてが最大区間面積問題, および最大正方区間面積問題の最適解 ($O(n)$ の解法) に到達した. ちなみに以前,「右隣の小さい数の表作成問題」を出題することなしに, 最大区間面積問題のみを学部1年生に出題したときは, 最適解を得たものは皆無であった. このことは,「右隣の小さい数の表作成問題」がプログラム作成問題として非常に重要であることを示唆している. なお, 表7において94年度と95年度の学生では, 第1回目の演習課題を最適解で答えた学生の割合に差が見られる. これは94年度の学生の中には非常に意欲的な学生が数名いて, 他の学生を引張ったためと推測される.

第2回の演習の前にP区間表の定義, 作成法 (図1), およびP区間表が $O(n)$ で作成できることの証明 (付録A.3) を講義した. すると第2回演習課題では, P区間表を使わない解法も含めてであるが, 半数以上の学生が最適解 ($O(n)$ の解法) を得ることができた. P区間表を使わない解法が多いのは, 第2回目の演習問題が比較的簡単なものであったためと思われる.

第3回演習課題の最大空部分行列問題は, 文献1) の演習問題の中で2番目に難しいとされている難問である. 累積表を利用するというヒントを与えはしたが, 80% 前後の学生が最適解 ($O(N)$ の解法) を得ることができた. また, より困難な最大矩形体積問題に対しても, 我々の予想を上回る正解率が得られた. なお, 94年度と95年度の学生の間に, 最大矩形体積問題を $O(N^{1.5})$ の解法 (3.7節の解法1) で答えた学生の割合に差が見られる (すなわち, 95年度は94年度の倍). これは, より良い解法, たとえば平均 $O(N \log N)$ の解法 (3.7節の解法2) に対してはAA (Aを2つ分) の評価を与えるという採点基準を公表したため, 94年度の学生が積極的に平均 $O(N \log N)$ のものを得る努力をし, かつ失敗したためである. 問題の難しさから判断して当然と考えられるが, 平均 $O(N \log N)$ の解法に到達した学生は皆無であった.

表8 P区間表の必要度
Table 8 P-sement tables are necessary.

絶対に必要である	あった方がよい	必要ない
75人	24人	1人

以上の結果を要約すると, 学生はP区間表による解法を十分に理解したうえで応用し, 難しい問題でも解決できた. したがって, P区間表による解法は多くの学生にとって, 理解も応用もしやすいものであるといえよう.

5.3 P区間表の好感度

新しいプログラミング技法は受け入れられやすい方が好ましい. P区間表が新しいプログラミング技法として必要であるか否かに関するアンケート調査結果を**表8**に示した. また, もしP区間表を知らない場合, 出題された演習問題を解くのにかかる時間はどの程度変化するかを予想させその結果を**表9**に示した. ちなみに, 一部の学生からは「P区間表を用いると一見異なる問題でも, 同一の問題に帰着でき, 解法を自然に理解できる」という意見が出た.

以上の結果を要約すると, 数列の区間や数平面上の矩形に関する問題を解決するうえで, P区間表は学生にとって必要かつ受け入れられやすいデータ構造であると考えられる.

5.4 P区間表の問題点と指導上の注意

P区間表を用いると簡単に効率の良い解法を得られるため, 初心者はそこで満足し, より効率の良い (計算量は同じでも, 定数がより小さい) 解法を見落とすことがある. 実際に最長の左最小区間問題 (左端が最小値となる区間の中で, 長さが最大なものを $O(n)$ 時間かつスペースで求める)[1]では, P区間表を用いるとその解法は自明である. しかし能率を考えるとより良い解法が他にある (P区間表を用いたときの時間計算量は $O(3n)$ であるのに対し, 別の解法では $O(2n)$). このような場合に, 効率の良い解法を見落とす学生が

表9 P区間表を知らない場合のプログラム作成時間（予想）
Table 9 Programming without using P-segment tables.

問題＼時間の予想	作成不能	より時間が掛かる	変わらない	短くなる
最大区間面積問題	57人（59%）	18人（18.5%）	18人（18.5%）	4人（4%）
最長素均衡区間問題	52人（54%）	19人（20%）	21人（22%）	4人（4%）
最長均衡区間問題	55人（57%）	18人（19%）	19人（20%）	4人（4%）
最大空部分行列問題	63人（65%）	22人（23%）	9人（9%）	3人（3%）
最大矩形体積問題	71人（75%）	13人（14%）	8人（8%）	3人（3%）

多く現れた．他の問題でもP区間表を用いない方が効率の良い解法を得られる場合がある．一般に，プログラムを作成した場合には他にもいくつかのアルゴリズムを考えて一番良いものを選ぶように指導するのが常識である．同様に，P区間表を指導する際にも，解法を得た後でも他により良い解法がないか調査する習慣をつけさせる必要がある．

もう1つの大きな問題は，与えられたプログラム作成問題の最適解がP区間表を用いて求まるか否か，現在のところ決定不能，すなわちやってみるまで分からないということである．たとえば，6章の最長平区間問題の一般の場合，最適解が$O(n)$か否かさえ目下不明である．しかし，まずP区間表により$O(n)$の解法が有るか否か試みることには価値がある．それにより我々の場合は，次章に示すように，特殊な場合に対する簡明な最適解を得ることができた．

6. 単調数列に対する最長平区間問題

P区間表は未知の問題解決にも有効な手段であることを示すために，この章では，新しい問題に対する最適解を区間表を用いて求める．問題は最長平区間問題[6]の単調数列に対する特殊な場合である．

数列における平区間とは，最大値と最小値の差がその長さ以下の区間である．問題は，与えられた数列に含まれるすべての平区間の中で最長なものの長さを求めることである．この問題の最適解は，筆者らの調査の範囲ではまだ知られていない．しかし我々は，数列が単調な場合の$O(n)$の解法を一種の最長区間表を用いて開発したので，その方法を以下に示す．その解法においては，最長区間表の作成自体は容易である．しかし，最長区間表という概念を用いることが解法の重要な手がかりとなっていることに注意されたい．

6.1 準　　備

最初にこの問題の解法に必要な関数，定義，定理を導入する．ただし，扱う数列は単調増加数列のみとする．

区間$[a,b]$の平坦度を表す関数$f(a,b)$を$f(a,b)=(b-a+1)-(x[b]-x[a])$とする．したがって区間$[a,b]$が平区間であるための必要十分条件は$f(a,b)\geq 0$である．

定義2 次の3つの条件を満たす区間$[a,b]$をaから始まる最長最平区間（longest flattest segment）と呼ぶ．これはaから始まる平坦度（すなわち，fの値）が最大な区間の中で最長のものである．

(1) $f(a,b)\geq 1$.
(2) $c\leq b$に対して，$f(a,c)\leq f(a,b)$.
(3) $b<d$に対して，$f(a,d)<f(a,b)$.

最長最平区間表をtとすると$[a,t[a]]$はaから始まる最長最平区間を表す．

区間に最長最平区間をつないで平区間にならなければ，それより後ろに伸びる平区間が存在しないことは，次の定理3により示すことができる．

定理3 $[a,b]$を区間，$[b+1,c]$を最長最平区間とする．このとき，$[a,c]$が平区間でないならば$b<d$なる任意のdに対し$[a,d]$は平区間ではない．

証明

$$\begin{aligned} f(a,d) &= f(a,b)-(x[b+1]-x[b])+f(b+1,d) \\ &\leq f(a,b)-(x[b+1]-x[b])+f(b+1,c) \\ &= f(a,c)<0. \end{aligned}$$

したがって，区間$[a,d]$は平区間ではない． □

6.2 単調数列に対する解法

まず，与えられた数列に対する最長最平区間表を下記6.3節の方法で$O(n)$時間およびスペースで作成する．与えられた数列の左端の点から右へ向かって走査を始めて，現在の点iに至るまでの最長平区間の長さをmaxとする．現在の点iを左端とする長さmaxの区間に$i+\max$からの最長最平区間をつなぐ．それが平区間ならば，maxを新しい平区間の長さで置き換える．そしてこの操作を平区間にならなくなるまで繰り返す．区間に最長最平区間をつないで平区間にならなければそれより後まで伸びる平区間は存在しない（定理3）ので，iを$i+1$とし，次の点から同様にして調べる．iから右に長さmaxの区間がとれなくなったら終了する．したがって最長平区間の長さmaxは，**図3**のアルゴリズムにより求めることができる．ただし，$flat(i,j)\equiv(f(i,j)\geq 0)$とする．図3中の

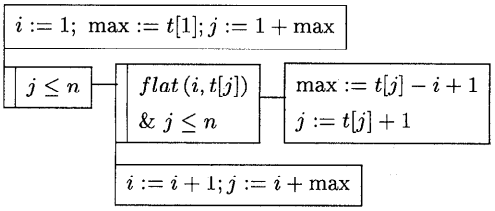


図3 単調区間に対する最長平区間問題の最適解法（ISO PAD[3]による記述）

Fig. 3 Optimal solution for the longest flat segment problem in the monotone case.

j はループを回るごとに1以上増加するので，ループが回る回数は n より小さい．したがって，図3のアルゴリズムは $O(n)$ である．

6.3 最長最平区間表の作成法

ここでは与えられた関係 $P(i,j)$ を，$[i,j]$ は i から右に伸びる平坦度（すなわち，関数 f の値）が最大の区間（最平区間）とし，最長最平区間表を作る方法を述べる．関数 f の定義より，$f(a, t[a+1]) = f(a,a) - (x[a+1] - x[a]) + f(a+1, t[a+1])$．また $f(a,a) - (x[a+1] - x[a])$ の値は，a から始まり $a+1$ 以降に伸びる区間にとって共通である．したがってその平坦度が1以上ならば区間 $[a, t[a+1]]$ が a から始まり，$a+1$ 以降まで伸びる区間のうちでは最大の平坦度を有する最長の区間であることが分かる．したがって各点 a から始まる最長最平区間を示す区間表 t は下記のように簡単に求められる（ただし，$t[n] = n$ とする）．

$$t[a] = \begin{cases} t[a+1] & : \text{if } f(a, t[a+1]) \geq 1 \\ a & : \text{otherwise.} \end{cases}$$

これは図1において，$Q(i,j) \equiv \text{true}$ とし，while ループを抜け出たところで $f(i, t[j])$ が1以上か否かに従って $t[i] = j$ かまたは $t[i] = i$ の設定を行うプログラムと同値である．

7. おわりに

(1) P 区間表と呼ばれる新しいデータ構造を提案し，それが $O(n)$ 時間およびスペースで作成できるための条件と作成法を示した．そして P 区間表を用いると数列の区間や数平面上の矩形に関する多くのプログラム作成問題を，統一的に効率良く解決できることを示した．

(2) P 区間表が数列の区間や数平面上の矩形に関するプログラム作成問題に対し，有効であり，学生に十分に受け入れられる方法であることを教育の現場で検証した．

(3) 現在までに $O(n)$ 解法の知られていない単調な数列に対する最長平区間問題の最適解を最長 P 区間表を用いて求めた．

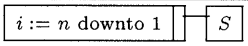
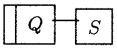
P 区間表の有効範囲を理論的により明確にすることと教育用教材の整備が今後の課題である．たとえば図1のアルゴリズムにより作成可能な P 区間表に対応する関係 P のサブクラスの解明は興味深い理論的問題であると考えている．

謝辞 早稲田大学理工学部筧捷彦教授は筆者らが本研究に取り組む機会となった興味深い問題提起をしてくださった．早稲田大学大学院理工学研究科遠藤貢一氏は P 区間表の有効性調査に協力してくださった．また早稲田大学理工学部情報学科3期および4期の学生諸君は，P 区間表を受け入れ積極的に利用してくださった．以上の諸氏に深謝します．最後に本稿を査読し，建設的なコメントをくださった複数の匿名の査読者に感謝します．

参考文献

1) Dijkstra, E.W. and Feijen, W.H.J.: *A Method of Programming*, Addison-Wesley, Reading, MA (1988).
2) 二村夏彦，二村良彦，筧　捷彦：対称ヒープの実現とその応用，情報処理学会アルゴリズム研究会 (1992).
3) 二村良彦：プログラム技法—PADによる構造化プログラミング，オーム社 (1984).
4) 二村良彦，白井千恵子，劉　咏梅，二村夏彦，筧　捷彦：P 区間表とそのプログラミング教育における効果，電子情報通信学会コンピューテーション研究会，pp.9–16 (1995).
5) 石畑　清：アルゴリズムとデータ構造，岩波書店 (1989).
6) 筧　捷彦：早稲田大学講義録プログラミング演習A (1991).
7) 白井千恵子，二村良彦：プログラミングにおけるスタックパラダイムの問題点，日本ソフトウェア科学会第12回大会，pp.21–24 (1995).
8) 白井千恵子，二村良彦：プログラミング教育におけるスタックの扱い，情報処理学会第37回プログラミングシンポジウム，pp.47–53 (1996).
9) Swierstra, D.: 私信 (1996).
10) Swierstra, D. and de Moor, O.: Virtual Data Structures, *Formal Program Development*, LNCS, Vol.755, pp.355–371, Springer-Verlag (1993).
11) Van der Woude, J.: Rabbitcount:=Rabbitcount–1, *Mathematics of Program Construction*, LNCS, Vol.375, pp.409–420, Springer-Verlag (1989).
12) 米田信夫：私信 (1993).

表10 PADとpascal風言語との対応
Table 10 Meaning of PAD symbols.

PADの記号	名前	対応するpascalの構文
	連接	S1; S2;
	反復	for i := n downto 1 do S;
	反復	while Q do S;

付　　録

A.1 ISO8631 PADにおける記号の説明

PADにおける記号のうちで本稿で用いられるものだけについて，pascal風言語との対応を**表10**に示した．

A.2 最短均衡区間表作成法の正しさの証明

図1において $r=1$，$Q(i,j) \equiv (x[i] \neq x[j])$ と設定して得られるアルゴリズムをBとする．ここではBによって最短均衡区間表が正しく作成されることを証明する．まず下記の補題1を証明する．

補題1 $x[1]\ldots x[n]$ を長さ n の区間，そして $t[1]\ldots t[n]$ を x に対する最短均衡区間表とする．このとき，$[1,n]$ が均衡区間であるための必要十分条件は $1 \leq c < n$ なる c が存在して，$[1,n]$ は c 個の互いに重複しない最短均衡区間の連で表すことができる．すなわち，$h[i] = t[i]+1$ とおくと，$[1,n] = [1,t[1]][t[1]+1,t[t[1]+1]]\ldots = [1,h[1]-1][h[1],h^2[1]-1]\ldots[h^{c-1}[1],h^c[1]-1]$.

証明

十分条件であることは明らかであるので，必要条件であることを n に関する数学的帰納法を用いて証明する．

(1) $n=2$ のとき，$[1,2]$ 自身が1つの最短均衡区間である．

(2) $n>2$ のとき，$t[1]=n$ ならば，t の定義より $[1,n]$ が1つの最短均衡区間である．$2<t[1]<n$ とする．このとき，$[1,n]=[1,h[1]-1][h[1],n]$ である．$[h[1],n]$ は長さが n より短い均衡区間であるので，帰納法の仮定より，$1 \leq c < n-h[1]$ なる c に対して，$[h[1],n] = [h[1],h^2[1]-1]\ldots[h^{c-1}[1],h^c[1]-1]$. □

アルゴリズムBは，まず j を $i+1$ として $x[i]$ と $x[j]$ を比べる．それが等しくなければ止まる．等しい場合には $t[j]$ から指されている要素の右隣の要素を j として上と同様の操作を繰り返し，初めて $x[i] \neq x[j]$ となる点を探す．これによって i から右に延びている均衡区間を見付ける．したがって，アルゴリズムBの正しさを証明するためには次の定理4を証明すればよい．

定理4 アルゴリズムBにおいて，j が $i+1$ から区間表 t をたどって，初めて $x[i] \neq x[j]$ となる点であるならば，区間 $[i,j]$ が i から始まる最短の均衡区間である．またそのような j が存在しない，すなわち $j=n+1$ になるならば，i から始まる均衡区間は存在しない．

証明

i に関する数学的帰納法により示す．

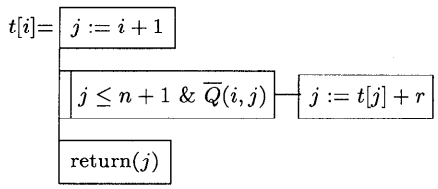
(1) $i=n$ のとき，$t[n]=n+1$ であるので，上記の j は存在しない．また，n から始まる均衡区間は存在しない．したがって定理は成立する．

(2) $1 \leq i < n$ のとき，$i < i' \leq n$ なる任意の i' について定理4の成立を仮定する．

(2.1) $j=i+1$ のとき，$x[i] \neq x[i+1]$ であるから明らかに $[i,j]$ は i から始まる最短均衡区間である．

(2.2) $n \geq j > i+1$ のとき，j の探し方より区間 $[i+1,j-1]$ は均衡区間をつないだものであるから，これも均衡区間である．したがって区間 $[i,j]$ は均衡区間である．ここで，区間 $[i,j]$ が i から始まる最短均衡区間でないと仮定する．すなわち，区間 $[i,j]$ が空でない2つの均衡区間 $[i,k]$ と $[k+1,j]$ に分割できると仮定する．また，$x[i]=0$ かつ $x[j]=1$ とする．

このとき，$[i,k]$ は均衡区間だから，$[i+1,k]$ は0の数が1の数よりも1つ少ない．しかし，$j>i+1$ より $x[i]=x[i+1]=0$ であるから，$i+1$ から始まる区間を左から走査すると，最初は0の数の方が多い．0の数の方が1つ少なくなるためには $i+1$ を右に k まで伸ばしてゆくとき，途中で均衡をなし，その直後に1が出現する必要がある．ところで，帰納法の仮定より，$i<i'$ なる i' に対しては $[i',t[i']]$ は i' から始まる最短均衡区間である．したがって補題1より，$i+1$ を左端とするすべての均衡区間は，最短均衡区間をつなげること，すなわち，$t[i+1]+1, t[t[i+1]+1]+1, \ldots$



ただし $n+1$ と nil は同じものと見なす.

図4 P 区間表作成のための再帰関数 $t[i]$ (ISO PAD [3] による記述)

Fig. 4 Recursive function $t[i]$ for making a P-segment table.

の連鎖をたどることによって,作成できる.したがって,上記の1は $i+1$ から始まる均衡区間の右隣の点であるから,$i+1$ から区間表をたどって訪れることができる.しかし,このことは,j が $i+1$ からたどって,初めて $x[i] \neq x[j]$ となる点であることに矛盾する.したがって,区間 $[i,j]$ は i から始まる最短均衡区間である.

(2.3) $j=n+1$ のとき,i から始まる均衡区間は存在しない.なぜならば,$[i,j']$ を $x[i]$ から始まる最短均衡区間とすれば,$x[i] \neq x[j']$ である.したがって,帰納法の仮定と補題1より,$x[j']$ はアルゴリズムBによって $n+1$ 以前にたどり着けなければならない. □

A.3 右 P 区間表作成の計算量に関する証明

証明

図4 の再帰関数 $t[i]$ により各点に対する右 P 区間表を作成する.ただし,$t[i]$ の計算中に使われる $t[j]$ $(j>i)$ の値は前もって計算済みとする.特に $n'>n$ に対して $t[n']$ には,関係 P に従って nil (すなわち $n+1$) を初期設定しておく.

また必要に応じて $x[n+1]$ 以降の値も設定しておくものとする.$r \geq 0$ である.図4の関数の計算量が図1のアルゴリズムと同じであることは明らかであろう.

まず,$h[i]=t[i]+r$ とおく.また $t[i]$ の値を $i=n-1,\ldots,1$ の順で計算する際に各 i について実際に実行される $\overline{Q}$ の回数を $ct(i)$ で表す.さらに $cands(i)$ を次のように定義する.これは関数 $t[i]$ 実行前の状態において,$i+1$ および $t[i+1]+r, t[t[i+1]+r]+r, \ldots,$ nil までの連鎖,すなわち $h^0[i+1], h^1[i+1], h^2[i+1], \ldots,$ nil をたどって到達できる要素の個数 (nil は含まない) である.この値は $\overline{Q}$ の実行回数の上限を示す.

$t[n+1]$ からの連鎖はないので $cands(n) \leq 1$ である.$t[i]$ の計算は,まず $\overline{Q}(i,i+1)$ を調べそれから $t[i+1]+r$ から nil までの連鎖をたどるが,その途中で必ず止まる.したがって式(1)が成立する.

$$ct(i) \leq cands(i). \tag{1}$$

また,$t[i]=h^{(ct(i)-1)}[i+1]$ より下記の式(2)が成立する.

$$\begin{aligned} cands(i-1) &= 1 + cands(i) - (ct(i)-1) \\ &= 2 + cands(i) - ct(i). \end{aligned} \tag{2}$$

これは,$t[i]$ が求められたことによりその計算の途中の連鎖に現れた $h[j]$ が $i-1$ から始まる最短 P 区間の右端の候補から除外されたことを意味する.上式(2)の i について,2から n までの和をとると,

$$\begin{aligned} &\sum_{i=2}^{n} cands(i-1) \\ &\quad = 2(n-1) + \sum_{i=2}^{n} cands(i) - \sum_{i=2}^{n} ct(i) \end{aligned} \tag{3}$$

となる.したがって

$$\begin{aligned} &\sum_{i=2}^{n} cands(i-1) - \sum_{i=2}^{n-1} cands(i) \\ &\quad = 2(n-1) + cands(n) - \sum_{i=2}^{n} ct(i). \end{aligned} \tag{4}$$

上式を整理すると

$$cands(1) \leq 2n - 1 - \sum_{i=2}^{n} ct(i).$$

上式(1)より,

$$ct(1) \leq cands(1) \leq 2n - 1 - \sum_{i=2}^{n} ct(i).$$

$$\text{ゆえに} \sum_{i=1}^{n} ct(i) \leq 2n-1.$$

すなわち $O(n)$ である.ちなみに $ct(i)$ の amortized time bound は2である.よって P 区間表作成の計算量は $O(n)$ 時間かつスペースであることが示された. □

(平成8年6月28日受付)
(平成9年4月3日採録)

二村 良彦（正会員）

1965年北海道大学理学部数学科卒業．1972年Harvard大学応用数学科大学院修士課程修了．工学博士．1965年日立製作所入社．日立製作所中央研究所主任研究員および基礎研究所主管研究員を経て，1991年より早稲田大学理工学部情報学科教授．その間，Uppsala大学計算機科学科Guest Professor（1985～1986年）およびHarvard大学計算機科学科Visiting Scholar（1988～1989年）．計算と論理の関係，プログラムの生産性向上，アルゴリズムの教育，設計，解析，評価などに興味を持つ．主要業績：Futamura Projectionsの発見（1971年），プログラム技法PADの開発とISO化（1979～1986：ISO8631），一般部分計算法の発見（1987年）と米国特許化（United States Patent No.5241678, 1993年8月31日成立）．日本ソフトウェア科学会，ACM各会員．

白井千恵子（学生会員）

1971年生．1995年早稲田大学理工学部情報学科卒業．現在，同大学院理工学研究科博士課程在学中．日本学術振興会特別研究員．アルゴリズムの設計・解析，プログラミングの方法論に興味を持つ．

劉 咏梅

1992年武漢大学計算機科学科修士課程修了，1992年より武漢大学ソフトウェア工学研究所講師．その間1994年早稲田大学理工学部訪問研究員．計算機科学の理論的側面に興味を持つ．特に，部分計算（評価），プログラミング技法の分野で仕事をしている．

二村 夏彦

1992年早稲田大学理工学部数学科卒業．1996年Syracuse大学計算機科学科修士課程修了．現在同大学博士課程在学中．確率的アルゴリズム，並列アルゴリズム，遺伝的アルゴリズム，アルゴリズムの複雑さ，計算理論などアルゴリズム全般に興味を持つ．