

# 効率的な 2 分木の構成法についての考察

2018SE041 松野 宏典

指導教員: 横山 哲郎

## 1 はじめに

2 分木は非線形的な構造をもつ最も基本的で古くから研究が行われてきたデータ構造である。互いに異なるラベルがつけられた 2 分木の前順走査と間順走査から元の 2 分木がその節数に対して線形の回数の要素比較で一意に作れることが少なくとも半世紀前から知られており最近まで研究されている[1-12]。この問題は前順走査で訪ねられた節に昇順に 1 から順に自然数が付与された、節数 $n$ の 2 分木の間順走査から元の 2 分木を作る問題に言い換えることができる。本稿では言い換えたこの問題を対象とする。

我々の知る範囲において、最良比較回数 $2n - 1$ と最悪比較回数 $3n - 2$ をもつ Mäkinen のアルゴリズム (以下, アルゴリズム M) [6]が, 報告されているアルゴリズムの中で, この問題を解く最も効率の良いものである。これに対して, 最近, Yokoyama と Glück によって考案されたアルゴリズム (以下, アルゴリズム YG) は, 最良比較回数が $n + 2$ , 最悪比較回数が $2n + \lfloor n/2 \rfloor$ であると主張されている[11]。

我々は, アルゴリズム A の正当性を示すことと解析をすることを目的とする。3 以下の節数のときに文献[12]で報告されているアルゴリズムについて正当性と比較回数の正しさを確かめることを研究課題とする。

## 2 2 分木

2 分木は, 根と呼ばれる特別な節を一つもち, 高々一つの左部分木と高々一つの右部分木をもつ。ここで, 左部分木と右部分木も 2 分木である。本稿では節はラベルをもたないとする。

2 分木の根を調べて再帰的に左部分木を調べて右部分木を調べる順序を前順という。再帰的に左部分木を調べて 2 分木の根を調べて右部分木を調べる順序を間順という。

図 1 に節数 3 以下の 2 分木を全列挙する。各 2 分木の下には対応する前順走査を示す。

## 3 アルゴリズム M とアルゴリズム YG

アルゴリズム M をエラー! 参照元が見つかりません。に, アルゴリズム YG を図 2 の流れ図として示す。アルゴリズムは端子 Begin から開

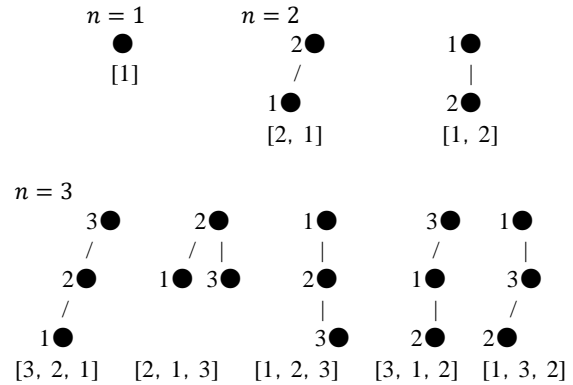


図 1 節数 3 以下の 2 分木と前順走査

始して, 端子 End で終了する。開始時には配列 **preorder** に前順走査で訪ねられた節に昇順に 1 から順に自然数が付与された, 節数 $n$ の 2 分木の間順走査が保存されている。配列の添え字として  $i$  を用いる。ラベルと節を格納するのにそれぞれスタックを用いる。

それぞれの処理と判断は次の通りである。

処理 A: **preorder**[ $n$ ]= $N+1$  とする。( $N = n + 1$ )

処理 A:  $i=1$  とする。節を作って節スタックにプッシュする。ラベルスタックにスタックのボトムであることを示すマーカをプッシュする。さらに根を作って節スタックにプッシュし, **preorder**[0] をラベルスタックにプッシュする。

処理 B: トップの節の左の子を作成し, 節スタックにプッシュする。また, 注目しているラベル **preorder**[ $i$ ] をラベルスタックにプッシュする。

処理 C: 節スタックからポップした節, すなわち直前に注目していた節を **prev** に格納する。また, ラベルスタックをポップする。

処理 D: 直前に注目していた節 **prev** の右の子を作って, 節スタックにプッシュする。また, 注目しているラベル **preorder**[ $i$ ] をラベルスタックにプッシュする。

処理 E:  $i$  を増分する。すなわち, 注目するラベルを前順走査のひとつ後ろの要素にする。

判断 $\alpha$ :  $i < n$  であるか, すなわち, 与えられた前順走査を全て処理したかの判断をする。

判断 $\beta$ :  $\text{preorder}[i]$  がラベルスタックにおいてトップのデータの葉の数より小さいかの判断をする。

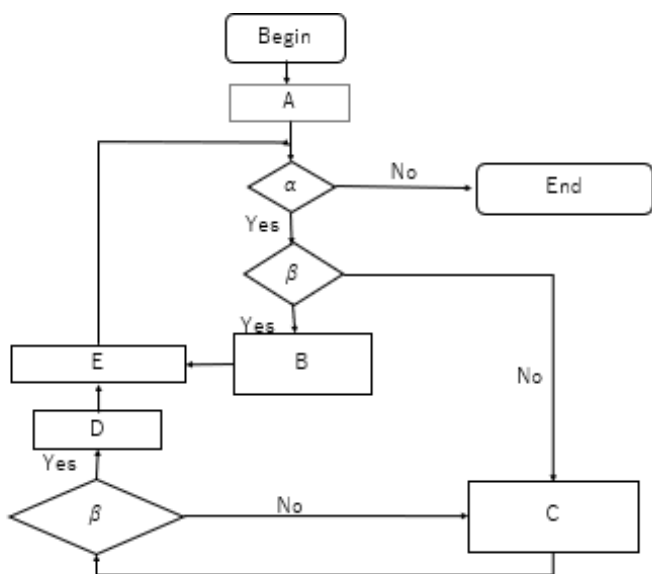


図 1 アルゴリズム M

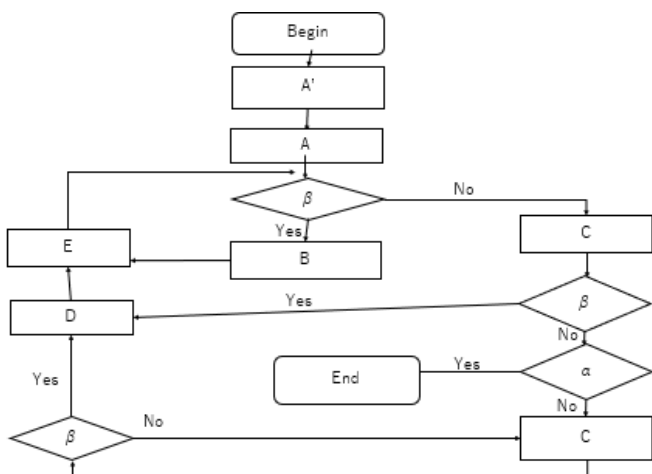


図 2 アルゴリズム YG

#### 4 アルゴリズム M とアルゴリズム YG の解析

図 1 に示した前順走査から対応する 2 分木が作られることを確かめた。ここで  $M_n$  と  $YG_n$  は節数  $n$  の前順走査を入力したときにそれぞれアルゴリズム M と YG を実行したときの比較回数を表す。また、必要な比較の回数は表 1 のようであった。これらの結果は文献[11]のものとも一致した。以上のように、3 以下の節数のときにアルゴリズム YG の正当性と比較回数の正しさを確かめられた。

表 1 節数 3 以下の場合の M と YG の比較回数

節数 $n$	2 分木の 総数	比較回数		$\frac{YG_n}{M_n}$
		$M_n$	$YG_n$	
1	1	1	3	3.0000
2	2	7	9	1.2857
3	5	31	32	1.0322

#### 参考文献

- [1] Arne Andersson and Svante Carlsson. Construction of a tree from its traversals in optimal time and space. *Information Processing Letters*, 34(1):21–25, 1990.
- [2] Ben Johnsen. Generating binary trees with uniform probability. *BIT Numerical Mathematics*, 31(1):15–31, 1991.
- [3] David Gries and Jan L. van de Snepscheut. Formal development programs and proofs. In Edsger W. Dijkstra, editor, *Formal Development of Programs and Proofs*, pp. 37–42. Addison-Wesley, 1990.
- [4] Donald E. Knuth. *The Art of Computer Programming*, Volume 1: Fundamental Algorithms, Addison-Wesley, 1st edition, 1968.
- [5] Erkki Mäkinen. Constructing a binary tree from its traversals. *BIT Numerical Mathematics*, 29(3):572–575, 1989.
- [6] Erkki Mäkinen. Constructing a binary tree efficiently from its traversals. *International Journal of Computer Mathematics*, 75(2):143–147, 2000.
- [7] Jan L. A. van de Snepscheut. Inversion of a recursive tree traversal. *Information Processing Letters*, 39(5):265–267, 1991.
- [8] Limin Xiang, Armin Lawi, and Kazuo Ushijima. On constructing a binary tree from its traversals. Technical report, Faculty of Information Science and Electrical Engineering, Kyushu University, 5(1):13–18, 2000.
- [9] Robert Glück and Tetsuo Yokoyama. Constructing a binary tree from its traversals by reversible recursion and iteration. *Information Processing Letters*, 147:32–37, 2019.
- [10] Shin-Cheng Mu and Richard Bird. Rebuilding a tree from its traversals: In Atsushi Ohori, editor, *Programming Languages and Systems*, pages 265–282. Springer, 2003.
- [11] Tetsuo Yokoyama, Robert Glück: A Note on Efficient Binary Tree Construction (Preprint), 2021.
- [12] Wei Chen and Jan Tijmen Udding. Program inversion: More than fun! *Science of Computer Programming*, 15(1):1–13, 1990.