アルゴリズム及び演習 第 9 回演習解答

小野 孝男[*]

2007 年 6 月 18 日

1. 最初の 3 個を順に 2 分探索木に入れると図 1 (a)〜(c) のようになる. 最終的には図 2 の 2 分探索木が得られる.

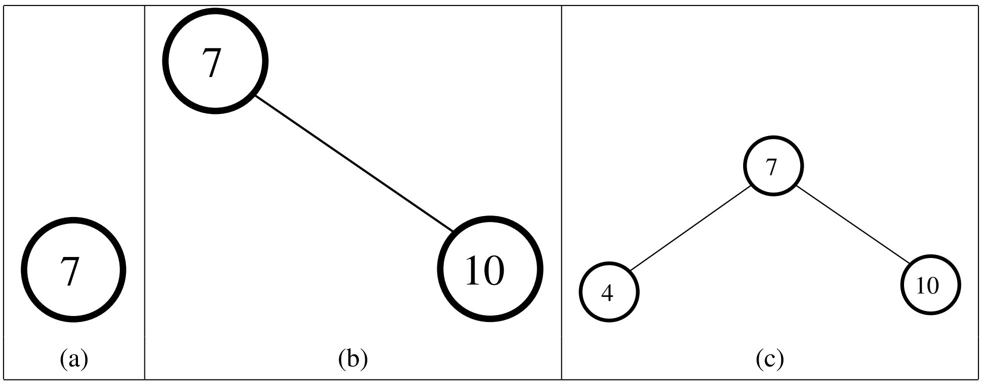

図 1　2 分探索木に最初の 3 個のデータを順に入れた例

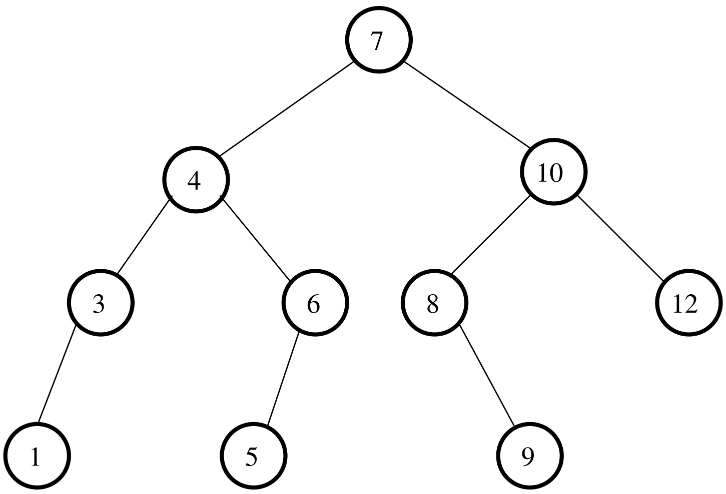

図 2　最終的にできる 2 分探索木

2. 単純には, 次の 2 つのステップを実行すればよい:

[*] ono@is.nagoya-u.ac.jp

(a) T を探索し, x のあるべき場所を見付ける. ここで, 「x より大きい要素」から「x より小さい要素」に移る (あるいはその逆) ときにはそのリンクを削除して元の頂点を適切なスタックに積む.

(b) 木の葉に辿りついたら, スタックからポップしつつ木を修復する.

このアルゴリズムは木を根から (適切な) 葉まで辿り, その後逆に根まで戻るという動作をするため実行時間が $O(h)$ となることはほぼ明らかだろう.

再帰的アルゴリズムを使えばスタックを管理しなくてよいので便利である. そこで, この問題を再帰的に解くアルゴリズム SplitTree(r, x) を示す. ここで r は対象とする木 T の根である. SplitTreeL 及び SplitTreeR は補助手続きであり, これらが互いに再帰呼び出しを行うことで木の分解と修復を行う.

```
procedure SplitTree(r, x)
    if r = nil then return (nil, nil)
    else if r.data > x then return SplitTreeR(r, x)
    elsereturn SplitTreeL(r, x)
    end if
end procedure
procedure SplitTreeR(r, x)
    p ← r
    while p ≠ nil&p.data > x do
        p ← p.left
    end while
    if p = nil then return (nil, r)
    end if
    q ← p.parent
    (r_L, r_R) ← SplitTreeL(p, x)
    q.left ← r_R
    return (r_L, q)
end procedure
procedure SplitTreeL(r, x)
    p ← r
    while p ≠ nil&p.data < x do
        p ← p.right
    end while
    if p = nil then return (r, nil)
    end if
    q ← p.parent
    (r_L, r_R) ← SplitTreeR(p, x)
    q.right ← r_L
    return (q, r_R)
end procedure
```

3. 以下の通りである:

(a) n 個の頂点からなる 2 分探索木の根を r とすると, r の左右の部分木はいずれも 2 分探索木であり, その頂点数の合計は $n-1$ 個である. つまり, 左部分木の頂点数を k とすると右部分木の頂点数は $n-1-k$ 個である. このような頂点の割当に対し, 左部分木には b_k 通り, 右部分木には b_{n-1-k} 通り考えられるので, 合計で $b_k b_{n-1-k}$ 通りとなる ($k=0$ 及び $k=n-1$ のときにも, $b_0=1$ と定義したの

でこれは正しい). ここで左部分木の頂点数 k は 0 以上 $n-1$ 以下なので

$$b_n = \sum_{k=0}^{n-1} b_k b_{n-1-k}$$

である.

（b）漸化式

$$b_n = \sum_{k=0}^{n-1} b_k b_{n-1-k}$$

の両辺に x^n を掛けて $n = 0$〜∞ の和をとる. 左辺は

$$\sum_{n=0}^{\infty} b_n x^n = B(x)$$

である. 右辺は

$$\begin{aligned}\sum_{n=0}^{\infty}\left(\sum_{k=0}^{n-1} b_k b_{n-1-k}\right)x^n &= 1 + \sum_{n=1}^{\infty} x\left(\sum_{k=0}^{n-1} b_k b_{n-1-k}\right)x^{n-1} \\ &= 1 + x\sum_{n=1}^{\infty}\sum_{k=0}^{n-1}(b_k x^k)(b_{n-1-k}x^{n-1-k})\end{aligned}$$

となるが, 右辺は和の順序を交換すると

$$1 + x\left(\sum_{n=0}^{\infty} b_n x^n\right)^2 = 1 + xB(x)^2$$

となる. つまり

$$B(x) = 1 + xB(x)^2$$

である.

（c）$\sqrt{1+x}$ の McLaurin 展開を用いて $\sqrt{1-4x}$ を展開する.

$$\begin{aligned}&\sqrt{1-4x} \\ &= 1 + \frac{1}{2}(-4x) - \left(\frac{1}{2}\right)^2\frac{(-4x)^2}{2!} + \left(\frac{1}{2}\right)^2\left(\frac{3}{2}\right)\frac{(-4x)^3}{3!} - \cdots + (-1)^{k-1}\left(\frac{1}{2}\right)^2\left(\frac{3}{2}\right)\cdots\left(\frac{2k-3}{2}\right)\frac{(-4x)^k}{k!} + \cdots \\ &= 1 - 2x - \frac{(2x)^2}{2!} - \frac{3(2x)^3}{3!} - \cdots - \frac{1\cdot 1\cdot 3\ldots(2k-3)(2x)^k}{k!} - \cdots \\ &= 1 - 2x - \frac{(2x)^2}{2!} - \frac{2\cdot 3\cdot 4x^3}{2\cdot 3!} - \cdots - \frac{2\cdot(2k-2)!x^k}{k!(k-1)!} - \cdots.\end{aligned}$$

従って $B(x) = (1 - \sqrt{1-4x})/(2x)$ における x^{n-1} の係数 b_{n-1} は

$$b_{n-1} = \frac{2\cdot(2n-2)!}{2\cdot n!(n-1)!} = \frac{(2n-2)!}{n\cdot[(n-1)!]^2} = \frac{1}{n}\binom{2n-2}{n-1}$$

となるので $b_n = \binom{2n}{n}/(n+1)$ である.