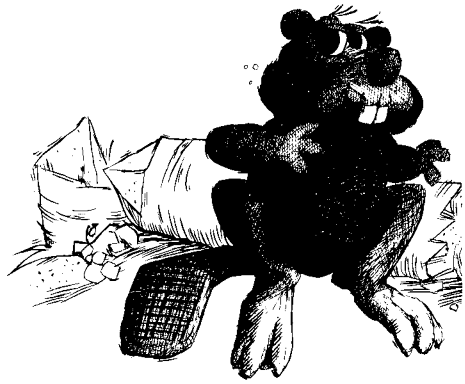
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



The Node Visit Cost of Brother Trees

Rolf Klein
Derick Wood

Data Structuring Group

CS-86-40

September, 1986

The Node Visit Cost of Brother Trees *

Rolf Klein † Derick Wood ‡

September 19, 1986

Abstract

The performance of a brother search tree depends on its shape; it can be measured by three basic cost measures: node visit cost, comparison cost, and space cost. The structure of brother trees that are optimal with respect to each of these cost measures is already known, as well as how to construct them in linear time. In this paper we investigate sharp bounds for the range that the node visit cost may take for a given size of tree. To this end we determine the structure of those brother trees which, for a given size N, have maximal (or pessimal) node visit cost. We derive a tight upper bound for the node visit cost of brother search trees which together with the lower bound obtained earlier yields the desired range estimation. Furthermore, we show that at least 11.6% of the internal nodes of a brother tree of maximal height are unary.

Key-words: brother trees, node visit cost, worst case cost

1 Introduction

For the implementation of dictionaries several balanced tree schemes are at the implementors disposal which allow arbitrary sequences of the dictionary operations *insert*, *delete* and *member* to be implemented in time $O(\log N)$, N being the number of keys currently present in the structure. Some of these schemes are: balancing the weight of subtrees as in the $BB[\alpha]$ trees of [3] and balancing the height of subtrees as in the AVL trees of [1] and the (a, b)-trees of [2]. The latter class also includes the brother (leaf-search) trees and brother search trees investigated in [5,6,7,8].

*This work was partially supported by Natural Sciences and Engineering Research Council of Canada Grant A-5692.

†Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, Postfach 6980, D-7000 Karlsruhe, West Germany

‡Data Structuring Group, Department of Computer Science, University of Waterloo, Ontario N2L 3G1, Canada

In order to select an appropriate balanced tree scheme for a specific application the implementor should know about the time and space requirements of the different schemes in some detail. At least he should have at hand good estimates for the constants that are hidden in the corresponding "big O"s. Very often a careful investigation of a given scheme is necessary to obtain precise estimates. This frequently leads to structural results that are interesting in their own right, because they help us to understand the intrinsic behaviour of our data structures and enrich the theory of trees.

Concerning brother trees, their storage utilization has been studied in [8] in both the worst case and the average case, and shown to be better than the memory utilization of $(2,3)$-trees. If, at the initialization phase of a dictionary, a set of N keys to be stored is given in lexicographical order and if member queries occur more frequently than insert or delete operations it is natural to ask for a brother search tree that can hold N keys and is optimal in some sense. Once such a tree is given the N keys can be placed in linear time by traversing the tree. This problem has been solved in [4] with respect to three different cost measures, the space cost, the comparison cost, and the node visit cost. For each of these cost measures the structure of the optimal brother tree for N keys has been characterized. Furthermore, algorithms have been presented that allow us to construct optimal brother trees in linear time.

In the present paper we continue the work begun in [4]. We start to investigate the general case when brother search trees are built by inserting the initial set of keys in arbitrary order (resp. dynamically altered by sequences of insert and delete operations). Here we are interested in the worst case behavior of our cost measures and in the structure of the brother search trees which attain maximal cost. In this paper we address the node visit cost. For each integer N we first characterize the structure of all brother search trees for N keys that have maximal node visit cost. This is done by determining their detailed profile. Then we derive a formula for the maximal node visit cost depending on N and give a tight upper bound for this function. Together with the lower bound obtained in [4] this gives the precise range the node visit cost of a brother tree can take. As a further consequence we show that at least 11.6% of the internal nodes of a brother tree of maximal height are unary.

2 Brother trees, detailed profiles and node visit cost

In this paragraph we recall the basic definitions that will be needed in the sequel. They can also be found in [4].

A *brother tree* is a rooted, oriented tree each of whose internal nodes

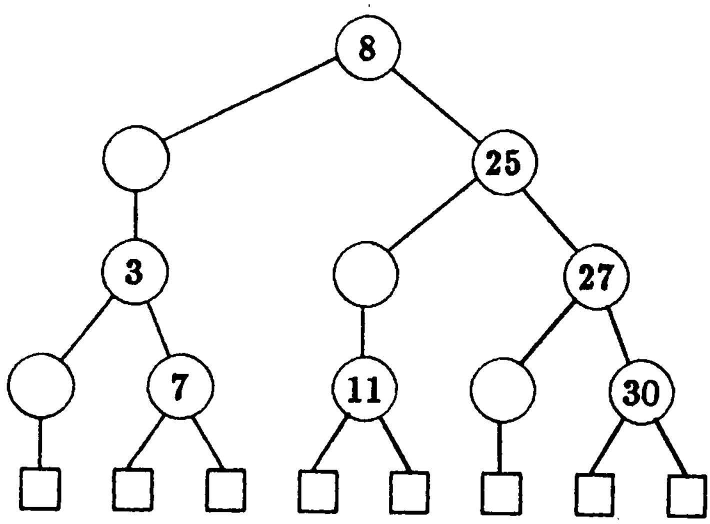


Figure 1: A brother search tree.

has either one or two sons. Each unary node must have a binary brother. All external nodes are at the same depht. As a consequence of the brother condition, the root of a brother tree must be binary.

There are two different methods of storing keys in a brother tree. First, we can store keys in ascending order in the external nodes and use the internal nodes for routing or separating values only. This leads to the class of *brother leaf search trees.* Second, we can associate one key to each internal binary node while the internal unary nodes and the external nodes remain empty. Here the keys are stored in inorder. The latter results in the class of *brother search trees* (also called *1-2 brother trees* in the literature). The two different methods correspond to the usual notions of external, resp., internal search trees. In this paper we are concerned only with brother search trees, so we often refer to them as, simply, brother trees.

Figure 1 shows an example of a brother search tree for the keys 3, 7, 8, 11, 25, 27, 30. In general, we call the number of internal binary nodes of a brother tree its *size.* Note that the number of external nodes of a brother tree is always by 1 greater than its size.

We count the *level* of nodes starting with level 0 at the root. So the external nodes of the tree shown in Figure 1 are on level 4. Therefore, the *height* of this tree equals 4.

If T is a binary tree with external nodes on level h only then the *detailed profile* $\Delta(T)$ of T is the sequence

$$< \omega_0, \beta_0 >, < \omega_1, \beta_1 >, \ldots, < \omega_h, \beta_h >$$

of ordered pairs $< \omega_i, \beta_i >$ where

$$\begin{aligned}\omega_i &= \text{number of unary nodes on level } i \\ \beta_i &= \text{number of binary nodes on level } i\end{aligned}$$

for $0 \leq i \leq h-1$ and

$$\begin{aligned}\omega_h &= 0 \\ \beta_h &= \text{number of external nodes} = 1 + \sum_{i=0}^{h-1} \beta_i\end{aligned}$$

Thus, all the external nodes are binary by definition. Furthermore, we define $\nu_m = \omega_m + \beta_m$ to be the number of nodes at level m. The tree T in Figure 1 has detailed profile $\Delta(T) = < 0,1 >, < 1,1 >, < 1,2 >, < 2,3 >, < 0,8 >$, for example. The notion of detailed profile turned out to be an appropriate tool for the analysis of brother trees in [4] already.

Let T be a brother tree of height h and with detailed profile $\Delta(T) = < \omega_0, \beta_0 >, < \omega_1, \beta_1 >, \ldots, < \omega_h, \beta_h >$. Then the *node visit cost* of T is defined by

$$NVCOST(T) = \sum_{i=0}^{h-1} (i+1)\beta_i$$

We have $NVCOST(T) = 21$ for the tree T in Figure 1. If $N = \beta_h - 1$ is the size of T then

$$\frac{1}{N} \cdot NVCOST(T)$$

is just the average number of node visits per access for a brother search tree, because $i+1$ nodes must be visited in order to access a node on level i, and keys are associated only to internal binary nodes. In a PASCAL implementation of a brother search tree, *NVCOST* is a measure for the number of pointers that must be traced to access the keys. The other constituent part of the time complexity is the number of key comparisons which are necessary to access the keys, that is, the *comparison cost* $NVCOST(T)$ (see [4] for details). Here we are investigating the worst case behavior of $NVCOST(T)$.

The significance of detailed profiles is immediate.

(2.1) Proposition: *The node visit cost $NVCOST(T)$ of a brother tree depends only on its detailed profile $\Delta(T)$.*

In order to investigate the node visit cost we will consider detailed profiles of brother trees rather than the trees themselves. For this purpose we must determine which sequences of ordered pairs of integers are detailed profiles of brother trees, and which are not.

(2.2) Proposition: *Let $h \geq 1$ and $< \omega_0, \beta_0 >, \ldots, < \omega_h, \beta_h > = \Delta$ be a sequence of ordered pairs of integers. Then Δ is detailed profile of a brother tree if and only if the following conditions are satisfied*

1. $\omega_i, \beta_i \geq 0, \; 0 \leq i \leq h$
2. $\omega_0 = 0, \; \beta_0 = 1$
3. $\omega_i + 2\beta_i = \nu_{i+1}, \; 0 \leq i \leq h-1$
4. $\omega_h = 0$
5. $\beta_i \geq \omega_{i+1}, \; 0 \leq i < h$

Proof: See [4]. □

Note that the ω_i can be computed recursively using (4) and (3), as soon as the β_i are known. Nevertheless, it is useful to include them into the detailed profiles.

There is one more result from [4] that will be needed in the sequel. It states that $NVCOST(T)$ is strictly increasing with the height $h(T)$.

(2.3) Proposition: *Let T and T' be brother trees which have the same number of external nodes. Assume that $h(T) < h(T')$ holds for the heights of T and T', respectively. Then $NVCOST(T) < NVCOST(T')$.*

Proof: See [4], Lemma 3.1. □

Note that the converse of (2.3) fails to hold. Figure 2 shows two brother trees which are both of height 3 but have *NVCOST* 12 and 11. The reason becomes clear by the formula

$$NVCOST(T) = (h-1)(N+1) + 1 - \sum_{i=0}^{h} \omega_i$$

which is obtained from (2.8) in [4] by applying the definition of the ν_i.

(2.4) Corollary: *A brother tree of size N is NVCOST pessimal if and only if*

1. *its height is maximal.*
2. *it has the minimal number of unary nodes under all competitors that satisfy 1.*

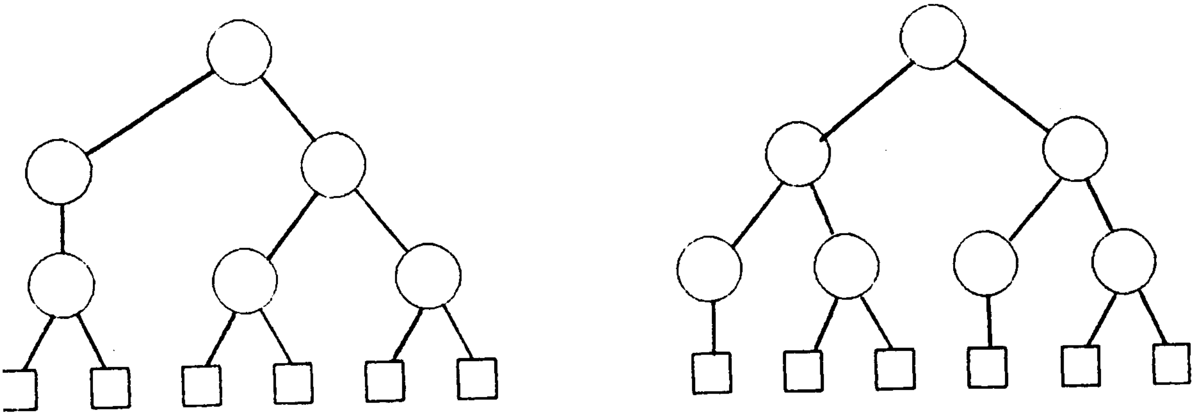


Figure 2: Two trees of the same height, but different *NVCOST*.

Intuitively, to maximize *NVCOST* for a given maximal height h means to have the unary nodes as close to the root as possible in order to give them as many as possible binary descendants. This minimizes the number of unary nodes.

3 Fibonacci trees

Our first goal is, for a given size N, to determine the detailed profile(s) of those brother trees which have maximal *NVCOST*. As a consequence of (2.3), these trees must be of maximal height. We start by stating this relationship the other way round.

(3.1) Lemma: *For given height $\hat{h} \geq 1$, the brother trees with minimal size have Fibonacci profile*

$$< 0, 1 >, < f_0, f_1 >, < f_1, f_2 >, \ldots, < f_{\hat{h}-2}, f_{\hat{h}-1} >, < 0, f_{\hat{h}+1} >$$

where $(f_i)_{i \geq 0}$ denotes the sequence of Fibonacci numbers $f_0 = f_1 = 1$, $f_{i+2} = f_{i+1} + f_i$, $i \geq 0$.

Proof: By induction on $\hat{h}$. Let T be a brother tree of height $\hat{h}$ and minimal size. For $\hat{h} = 1, 2$ the assertion is immediate. Assume $\hat{h} > 2$. Then T is, up to rotation, of one of the two shapes shown in Figure 3. Both T_l and T_r are brother trees and have minimal sizes with respect to their heights. By the induction hypothesis, their sizes are $f_{\hat{h}}$ and $f_{\hat{h}}$ in the first and $f_{\hat{h}-1}$ and $f_{\hat{h}}$ in the second case. Since $2f_{\hat{h}} > f_{\hat{h}-1} + f_{\hat{h}}$ the second case applies. Now the assertion follows by "adding" the Fibonacci profiles of T_l and T_r giving

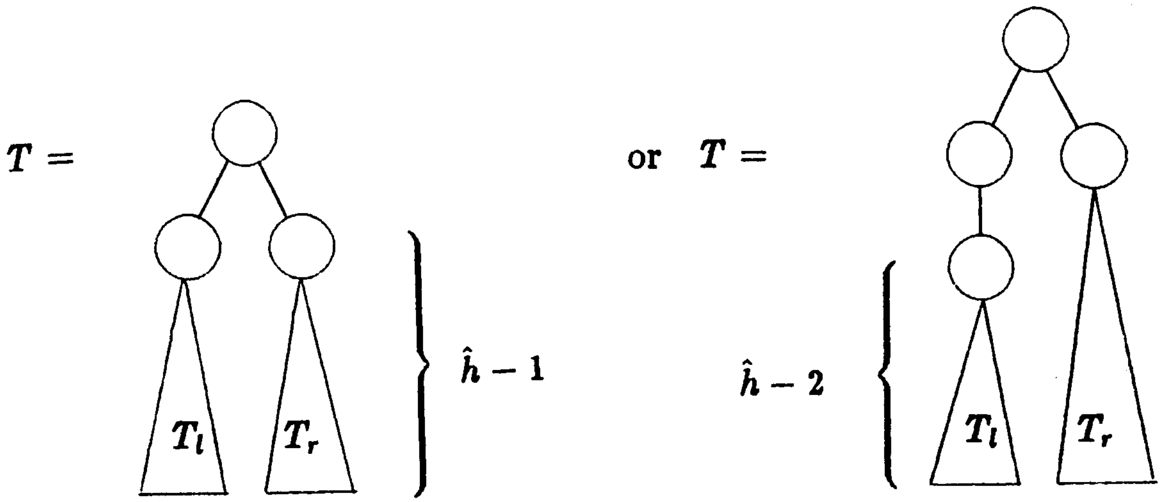


Figure 3: The trees used in the proof of Lemma 3.1.

$\Delta(T)$. □

In the sequel let $Fib(\hat{h})$ denote the set of all brother trees with detailed profile $< 0,1 >, < f_0, f_1 >, < f_1, f_2 >, \ldots, < f_{\hat{h}-2}, f_{\hat{h}-1} >, < 0, f_{\hat{h}+1} >$. Each tree in $Fib(\hat{h})$ has height $\hat{h}$ and $f_{\hat{h}+1}$ external nodes.

As a consequence of (3.1) we obtain

(3.2) Lemma: *Let T be a brother tree of height h and size ν. Then the following assertions hold for each integer $\hat{h}$.*

1. *If $\nu < f_{\hat{h}+1}$, then $h < \hat{h}$.*
2. *If $\nu = f_{\hat{h}+1}$, then $h \leq \hat{h}$.*
3. *If $\nu = f_{\hat{h}+1}$ and $h = \hat{h}$, then T is $Fib(\hat{h})$*

Proof: By contradiction, using (3.1). □

We conclude

(3.3) Corollary: *Let T be of size $f_{\hat{h}+1}$. Then $NVCOST(T)$ is maximal if and only if T is in $Fib(\hat{h})$.*

Proof: By (2) of (3.2), $h(T) \leq \hat{h}$, but the maximality of T implies $h(T) = \hat{h}$ by (2.3). Now the "only if" part follows from (3.2), 3, and the "if" part is

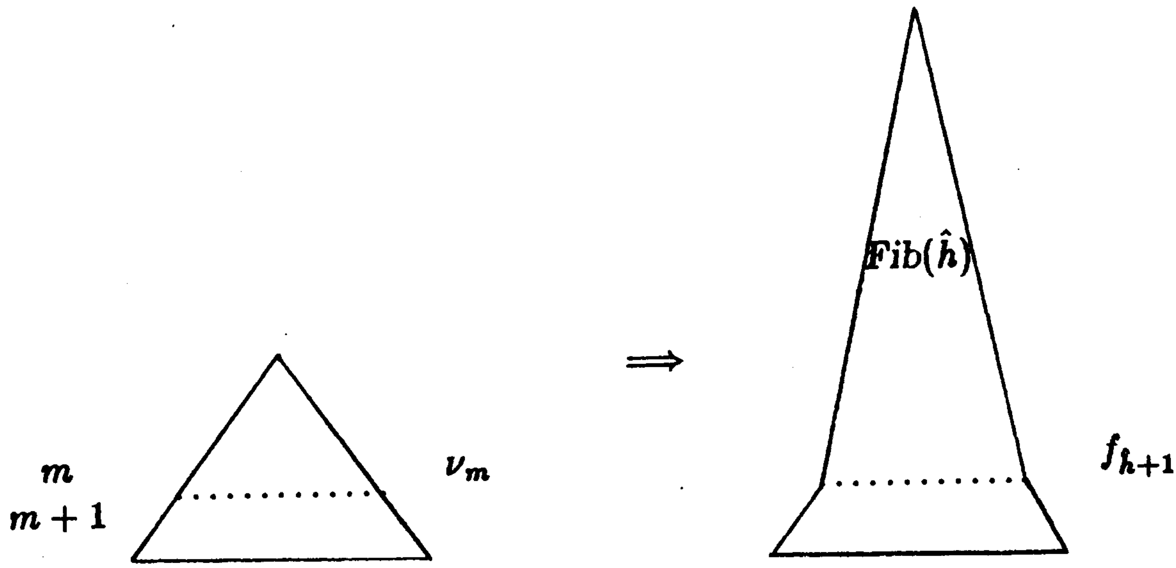


Figure 4: Fibonacci replacement.

a trivial consequence of (2.1). □

This solves our problem in the case that the size is a Fibonacci number; here the brother tree with maximal *NVCOST* is, up to rotations, a Fibonacci tree. But the Fibonacci numbers — and therefore the gaps between them, too — are growing exponentially. Thus, much remains to be done!

Fortunately, we can apply the same reasoning as above to the upper part of brother trees. The idea is as follows: Assume that the first m levels of a brother tree T could be replaced by a Fibonacci tree $Fib(\hat{h})$ (regarding its external nodes as being internal), without violating the brother tree properties of T, see Figure 4. If $\nu_m \leq f_{\hat{h}+1}$ then, by (3.2), the resulting tree is higher than T unless the replaced top of T is itself in $Fib(\hat{h})$. In all other cases $NVCOST(T)$ is increased, due to (2.3). This leads to a contradiction if we assume that T has maximal node visit cost.

First we describe the conditions under which the brother tree properties are not violated by this replacement.

(3.4) Lemma: *Let T be a brother tree with detailed profile $\Delta(T) = <\omega_0, \beta_0>, <\omega_1, \beta_1>, \ldots, <\omega_{m-1}, \beta_{m-1}>, <\omega_m, \beta_m>, <\omega_{m+1}, \beta_{m+1}>, \ldots, <\omega_h, \beta_h>$ with $1 \leq m < h$. Then for each integer $\hat{h}$ the following assertions are equivalent.*

A. $f_{\hat{h}+2} \leq \nu_{m+1} \leq 2f_{\hat{h}+1}$ and $\nu_{m+1} - f_{\hat{h}+1} \geq \omega_{m+1}$.

B. *The first m levels of T can be replaced by a tree in $Fib(\hat{h})$, that is, there are integers $\tilde{\omega}_{\hat{h}}$ and $\tilde{\beta}_{\hat{h}}$ such that*

$$\begin{aligned}\tilde{\Delta} \;=\; & <0,1>, <f_0, f_1>, <f_1, f_2>, \ldots, <f_{\hat{h}-2}, f_{\hat{h}-1}>, \\ & <\tilde{\omega}_{\hat{h}}, \tilde{\beta}_{\hat{h}}>, <\omega_{m+1}, \beta_{m+1}>, \ldots, <\omega_h, \beta_h>\end{aligned}$$

is the detailed profile of a brother tree $\tilde{T}$.

Then $\tilde{\omega}_{\hat{h}}$ and $\tilde{\beta}_{\hat{h}}$ are uniquely determined by the equations

$$\begin{aligned}\tilde{\omega}_{\hat{h}} &= 2f_{\hat{h}+1} - \nu_{m+1} \\ \tilde{\beta}_{\hat{h}} &= \nu_{m+1} - f_{\hat{h}+1}\end{aligned}$$

Proof: By (2.2) $\tilde{\Delta}$ is the detailed profile of a brother tree if and only if the following conditions are satisfied.

(a) $\tilde{\omega}_{\hat{h}} \geq 0$	(b) $\tilde{\beta}_{\hat{h}} \geq 0$
(c) $f_{\hat{h}-2} + 2f_{\hat{h}-1} = \tilde{\omega}_{\hat{h}} + \tilde{\beta}_{\hat{h}}$	(d) $\nu_{m+1} = \tilde{\omega}_{\hat{h}} + 2\tilde{\beta}_{\hat{h}}$
(e) $f_{\hat{h}-1} \geq \tilde{\omega}_{\hat{h}}$	(f) $\tilde{\beta}_{\hat{h}} \geq \omega_{m+1}$

Clearly (f) implies (b). The system of linear equations (c),(d) is equivalent to the representations of $\tilde{\omega}_{\hat{h}}$ and $\tilde{\beta}_{\hat{h}}$ in the theorem. Now the following equivalences hold.

$$\begin{aligned}\text{(a)} \;&\Leftrightarrow\; \nu_{m+1} \leq 2f_{\hat{h}+1} \\ \text{(e)} \;&\Leftrightarrow\; f_{\hat{h}-1} \geq 2f_{\hat{h}+1} - \nu_{m+1} \\ &\Leftrightarrow\; f_{\hat{h}+2} \leq \nu_{m+1} \\ \text{(f)} \;&\Leftrightarrow\; \nu_{m+1} - f_{\hat{h}+1} \geq \omega_{m+1}\end{aligned}$$

□

(3.5) Remarks:
1. The number $\hat{h}$ is uniquely determined by the equation $f_{\hat{h}+2} \leq \nu_{m+1} \leq 2f_{\hat{h}+1}$; for we have $2f_{\hat{h}+1} < f_{\hat{h}+3}$.
2. The condition $\nu_{m+1} - f_{\hat{h}+1} \geq \omega_{m+1}$ is a consequence of the first inequality in A, if $\omega_{m+1} \leq 1$. For we have $\nu_{m+1} - f_{\hat{h}+1} > \nu_{m+1} - f_{\hat{h}+2} \geq 0$.

(3.6) Corollary: *Let T be a brother tree of height h that has maximal NVCOST with respect to its size. Assume that $f_{\hat{h}+2} \leq \nu_{m+1} \leq 2f_{\hat{h}+1}$, $\omega_{m+1} \leq 1$, and $\nu_m \leq f_{\hat{h}+1}$ hold for integers m and $\hat{h}$ satisfying $0 \leq m < h$ and $\hat{h} \geq 0$. Then*

1. $\hat{h} = m$,

2. $< \omega_i, \beta_i > = < f_{i-1}, f_i >, \quad 1 \leq i \leq m-1$, and

3. $\omega_m = 2f_{\hat{h}+1} - \nu_{m+1}$ and $\beta_m = \nu_{m+1} - f_{\hat{h}+1}$.

Proof: Lemma (3.4) applies because we have $\omega_{m+1} \leq 1$ (see (3.5), 2). By assumption, $\nu_m \leq f_{\hat{h}+1} \leq \tilde{\nu}_{\hat{h}}$. Since T has maximal *NVCOST* (1) and (2) must hold. Now (3) is a consequence of (3.4). □

(3.7) Corollary: *The assertion of* (3.6) *remains valid if we drop the assumption* $\nu_m \leq f_{\hat{h}+1}$ *and assume* $\omega_m \leq 1$ *instead.*

Proof: $\omega_m \leq 1$ implies $\nu_m = \left\lceil \frac{\nu_{m+1}}{2} \right\rceil \leq f_{\hat{h}+1}$. □

4 Investigating the worst case structures

Throughout this section T is a brother tree of height h and size N that has detailed profile

$$\Delta(T) = < \omega_0, \beta_0 >, \ldots, < \omega_h, \beta_h >, \qquad \beta_h = N + 1$$

In the previous section we have shown that, if *NVCOST*(T) is maximal, the upper levels of T have Fibonacci profile if certain conditions are fulfilled. Now we prove a statement that draws a similar conclusion but presupposes different conditions. Both results together will lead to the desired structure theorem in Section 5.

First, we describe a technique for increasing the *NVCOST* of a brother tree by transformations of the type shown in Figure 5. Here the binary node 1 is moved down from level $m+1$ to level $m+2$, thereby increasing *NVCOST*(T) by 1. This transformation is the inverse of the technique that has been used in Lemma 3.2, [4], in order to *minimize* *NVCOST*(T).

(4.1) Lemma: *If* $\omega_{m+2} \geq 2$ *and* $\omega_{m+1} < \beta_m$, *then* *NVCOST*(T) *can be increased.*

Proof: We have to show that

$$< \omega_0, \beta_0 >, \ldots, < \omega_m, \beta_m >, < \omega_{m+1} + 1, \beta_{m+1} - 1 >,$$

$$< \omega_{m+2} - 2, \beta_{m+2} + 1 >, < \omega_{m+3}, \beta_{m+3} >, \ldots, < \omega_h, \beta_h >$$

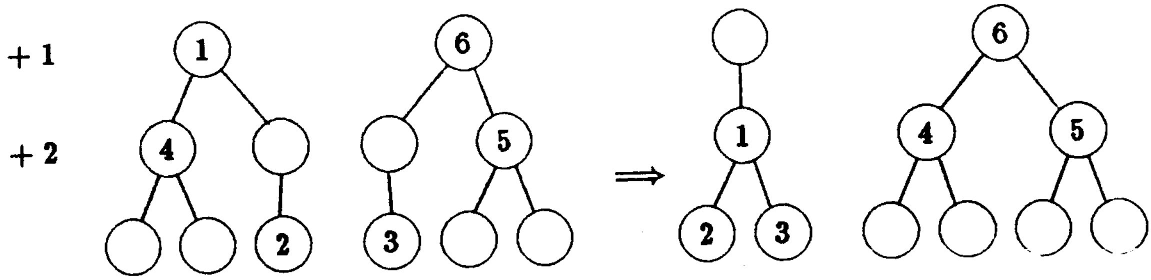


Figure 5: An increasing $NVCOST$ transformation.

is the detailed profile of a brother tree. Applying (2.2) we see that all conditions are fulfilled because $\Delta(T)$ is a valid profile, except that $\omega_{m+2} - 2 \geq 0$ and $\beta_m \geq \omega_{m+1} + 1$. But the latter hold by assumption. □

As an important consequence we get

(4.2) Lemma: *Assume that NVCOST(T) is maximal. Let $u \geq 0$ be maximal such that $\omega_{u+2} \geq 2$ (if such a u exists). Then*

$$< \omega_i, \beta_i > = < f_{i-1}, f_i >, \qquad i = 1, \ldots, u+1$$

Proof: Since $NVCOST(T)$ is maximal, (4.1) yields

$$(*) \qquad \omega_{m+2} \geq 2 \Rightarrow \omega_{m+1} = \beta_m$$

because $\omega_{m+1} \leq \beta_m$ holds by the brother tree Property (5) in (2.2). But in each brother tree we have $\beta_m \geq 2$ as long as $m \geq 2$. Thus repeated application of $(*)$ gives

$$\omega_{u+1} = \beta_u,\ \omega_u = \beta_{u-1}, \ldots, \omega_3 = \beta_2,\ \omega_2 = \beta_1$$

At this point the implication chain ends because $\beta_1 = 1$. For, if $\omega_2 = \beta_1 \geq 2$, then the next step would give $\omega_1 = \beta_0 = 1$; but $\nu_1 = \beta_1 + \omega_1 \geq 3$ is impossible. Now the assertion follows by induction on i:

$$< \omega_1, \beta_1 > = < 1, 1 > = < f_0, f_1 >$$

because $\omega_1 = \nu_1 - \beta_1 = 2 - 1 = 1$, $< \omega_{i+1}, \beta_{i+1} > = < \beta_i, \omega_i + \beta_i > = < f_i, f_{i+1} >$ because we have $\beta_{i+1} = \omega_i + 2\beta_i - \omega_{i+1}$ for all i and $\omega_{i+1} = \beta_i$ for $i = 1, \ldots, u$. □

By the maximality of u, the tree in (4.2) has at most one unary node on each of the levels $u+3, \ldots, h$. This determines the structure of these levels.

(4.3) Lemma: *Assume $\omega_j \leq 1$ for $j = q, \ldots, h$. Then*

$$\begin{aligned} < \omega_j, \beta_j > &= < \nu_{j+1} \; mod \; 2, \nu_{j+1} \; div \; 2 >, \qquad q \leq j \leq h-1 \\ \nu_j &= \left\lceil \frac{N+1}{2^{h-j}} \right\rceil, \quad q \leq j \leq h \end{aligned}$$

Proof: We have $\nu_{j+1} = 2\beta_j + \omega_j$ with $\omega_j \in \{0,1\}$. Clearly, $\nu_h = \left\lceil \frac{N+1}{2^0} \right\rceil$. Now

$$\begin{aligned} \nu_{j-1} &= \omega_{j-1} + \beta_{j-1} \\ &= \nu_j \; mod \; 2 + \nu_j \; div \; 2 \\ &= \left\lceil \frac{\nu_j}{2} \right\rceil \\ &= \left\lceil \frac{\left\lceil \frac{N+1}{2^{h-j}} \right\rceil}{2} \right\rceil \quad \text{by the induction hypothesis,} \\ &= \left\lceil \frac{N+1}{2^{h-j+1}} \right\rceil \end{aligned}$$

□

Let us summarize: If *NVCOST*(T) is maximal and if there is a u such that $\omega_{u+2} \geq 2$, then for the maximal u with this property, T looks like the tree shown in Figure 6. Here the levels 0 to $u+2$ have Fibonacci profile, and the lower part (levels $u+3, \ldots, h$) is as complete a binary tree as possible due to (4.3). By an application of (3.4) with $\hat{h} = u+2$ we can determine $< \omega_{u+2}, \beta_{u+2} >$, too, which gives us the complete detailed profile.

(4.4) Corollary: *Assume that NVCOST(T) is maximal and that there is an $u \geq 0$ such that $\omega_{u+2} \geq 2$. Let u be maximal. Then $\Delta(T)$ is determined by*

$$\begin{aligned} < \omega_0, \beta_0 > &= < 0, 1 > \\ < \omega_i, \beta_i > &= < f_{i-1}, f_i >, \qquad 1 \leq i \leq u+1 \\ < \omega_{u+2}, \beta_{u+2} > &= < 2f_{u+3} - \left\lceil \frac{N+1}{2^{h-(u+3)}} \right\rceil, \left\lceil \frac{N+1}{2^{h-(u+3)}} \right\rceil - f_{u+3} > \\ < \omega_j, \beta_j > &= < \nu_{j+1} \; mod \; 2, \nu_{j+1} \; div \; 2 >, \qquad u+3 \leq j \leq h-1 \\ < \omega_h, \beta_h > &= < 0, N+1 > \end{aligned}$$

Furthermore,

$$\nu_j = \left\lceil \frac{N+1}{2^{h-j}} \right\rceil, \quad u+3 \leq j \leq h$$

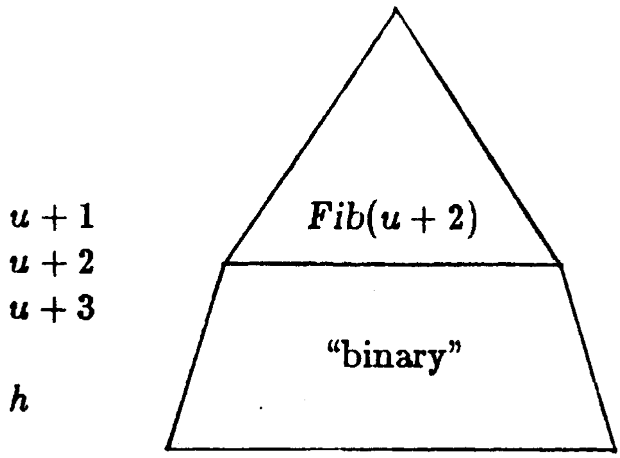


Figure 6: The general structure of maximal *NVCOST* trees.

$$f_{u+4} \;\leq\; \left\lceil \frac{N+1}{2^{h-(u+3)}} \right\rceil \leq 2f_{u+3}$$

5 The structure of brother trees with maximal NVCOST for given size

In view of the last lines in the previous section we define

(5.1) Definition: For each integer $N \geq 1$ let $\alpha(N)$ denote the smallest integer $k \geq 0$ such that there exists an integer $\hat{h} \geq 0$ which satisfies

$$f_{\hat{h}+2} \leq \left\lceil \frac{N+1}{2^k} \right\rceil \leq 2f_{\hat{h}+1}$$

Note that $\hat{h}$ is uniquely determined by k. We shall discuss the function α later. For the time being we need

(5.2) Proposition: *If T is a brother tree of height h and size N then $\alpha(N) \leq h - 1$.*

Proof: Since $N + 1 \leq 2^h$ we have $\left\lceil \frac{N+1}{2^h} \right\rceil = 1$. Thus, there is a number $k \leq h - 1$ satisfying

$$f_2 = 2 = \left\lceil \frac{N+1}{2^k} \right\rceil = 2f_1$$

□

Now we can prove the first part of our structure theorem.

(5.3.A) Theorem: *Given an integer $N \geq 1$, the NVCOST pessimal brother trees of size N are uniquely determined by the following detailed profile:*

$$
\begin{aligned}
<\omega_0, \beta_0> &= <0, 1> \\
<\omega_i, \beta_i> &= <f_{i-1}, f_i>, \quad 1 \leq i \leq \hat{h} - 1 \\
<\omega_{\hat{h}}, \beta_{\hat{h}}> &= <2f_{\hat{h}+1} - \left\lceil \frac{N+1}{2^k} \right\rceil, \left\lceil \frac{N+1}{2^k} \right\rceil - f_{\hat{h}+1}> \\
<\omega_j, \beta_j> &= <\nu_{j+1}\ mod\ 2,\ \nu_{j+1}\ div\ 2>, \quad \hat{h} + 1 \leq j \leq h - 1 \\
<\omega_h, \beta_h> &= <0, N+1>
\end{aligned}
$$

for $k = \alpha(N)$, $\hat{h}$ as defined in (5.1), and $h = \hat{h} + k + 1$.

Proof: Because of (2.1) it suffices to show: If T is a brother tree of size N and if $NVCOST(T)$ is maximal, then its detailed profile is as stated in the theorem. Let

$$\Delta(T) = <\omega_0, \beta_0>, \ldots, <\omega_{\tilde{h}}, \beta_{\tilde{h}}>$$

Case 1: $\omega_j \leq 1, \quad 0 \leq j \leq \tilde{h}$. By (4.3),

$$
\begin{aligned}
<\omega_j, \beta_j> &= <\nu_{j+1}\ mod\ 2,\ \nu_{j+1}\ div\ 2>, \quad 0 \leq j \leq \tilde{h} - 1 \\
\nu_j &= \left\lceil \frac{N+1}{2^{\tilde{h}-j}} \right\rceil, 0 \leq j \leq \tilde{h}
\end{aligned}
$$

From (5.2) we get $k = \alpha(N) \leq \tilde{h} - 1$. Thus, for $m := \tilde{h} - k - 1$ we have

$$f_{\hat{h}+2} \leq \nu_{m+1} = \left\lceil \frac{N+1}{2^k} \right\rceil \leq 2f_{\hat{h}+1}$$

by definition of α, and $\omega_m \leq 1$, $\omega_{m+1} \leq 1$ by assumption. Now the application of (3.7) yields

$$
\begin{aligned}
\tilde{h} - k - 1 &= m = \hat{h}, \\
<\omega_i, \beta_i> &= <f_{i-1}, f_i>, \ 1 \leq i \leq \hat{h} - 1 \\
<\omega_{\hat{h}}, \beta_{\hat{h}}> &= <2f_{\hat{h}+1} - \left\lceil \frac{N+1}{2^k} \right\rceil, \left\lceil \frac{N+1}{2^k} \right\rceil - f_{\hat{h}+1}>
\end{aligned}
$$

Therefore, $\tilde{h} = h$, and the assertion follows. (Note that $f_{\hat{h}-2} \leq 1$ as a consequence of the above equations. This is only possible if $\tilde{h} - k - 1 = \hat{h} \leq 3$; see Figure 7 below.)

Case 2: There is a $u \geq 0$ such that $\omega_{u+2} \geq 2$. Let u be maximal. Now the assertion of the theorem follows from (4.4), but with $u+2$ instead of $\hat{h}$ and $\tilde{h} - (u+3)$ instead of k. We have $k \leq \tilde{h} - (u+3)$ by the minimality of $k = \alpha(N)$. If $k = \tilde{h} - (u+3)$, then the last inequality of (4.4) implies $\hat{h} = u+2$, and the proof is complete. Now assume $k < \tilde{h} - (u+3)$. Let

$$\left\lceil \frac{N+1}{2^k} \right\rceil = \nu_{m+1}$$

for an integer $m \geq u+3$. Then $\omega_m \leq 1$, $\omega_{m+1} \leq 1$, (3.7) yields $\hat{h} = m$, and

$$\begin{aligned} < \omega_i, \beta_i > &= < f_{i-1}, f_i >, \ 1 \leq i \leq \hat{h} - 1 \\ < \omega_{\hat{h}}, \beta_{\hat{h}} > &= < 2f_{\hat{h}+1} - \left\lceil \frac{N+1}{2^k} \right\rceil, \left\lceil \frac{N+1}{2^k} \right\rceil - f_{\hat{h}+1} > \end{aligned}$$

By (4.4), the pairs $< \omega_{\hat{h}+1}, \beta_{\hat{h}+1} >, \ldots, < \omega_{\tilde{h}}, \beta_{\tilde{h}} >$ also have the values our theorem claims because $\hat{h} \geq u+3$. Note that $\hat{h} \geq u+4$ would imply $f_{\hat{h}-2} \leq 1$, a contradiction. Thus $\hat{h} = u+3$ holds, and from the equations

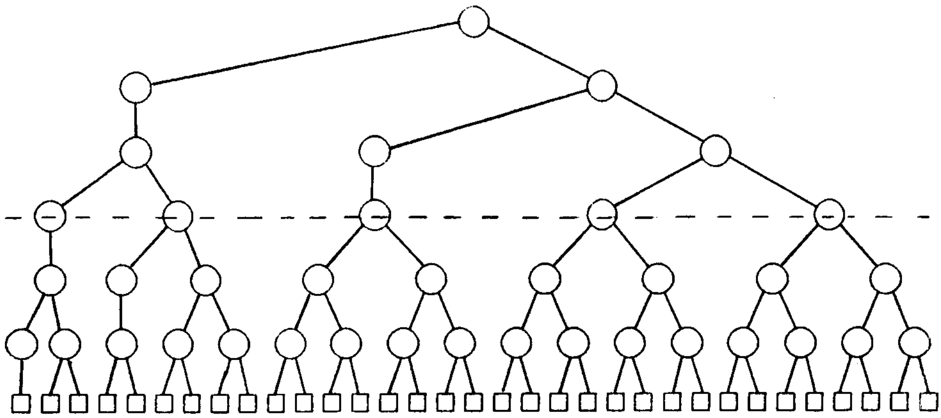
$$\begin{aligned} \left\lceil \frac{N+1}{2^k} \right\rceil &= \nu_{m+1} = \nu_{\hat{h}+1}, \\ \left\lceil \frac{N+1}{2^{\tilde{h}-\hat{h}}} \right\rceil &= \nu_{\hat{h}} \end{aligned}$$

we infer $\tilde{h} - \hat{h} = k+1$, which completes the proof. Figure 8 gives an example for the last case. □

Roughly speaking, this theorem has the following interpretation: To obtain a brother tree of size N that has maximal *NVCOST* proceed bottom-up as follows: If it is possible to obtain a Fibonacci number of nodes at the next level without violating the brother tree properties, then do so, and put the corresponding Fibonacci tree on top. Otherwise, halve the number of nodes by introducing at most one unary node at the next level and repeat this process.

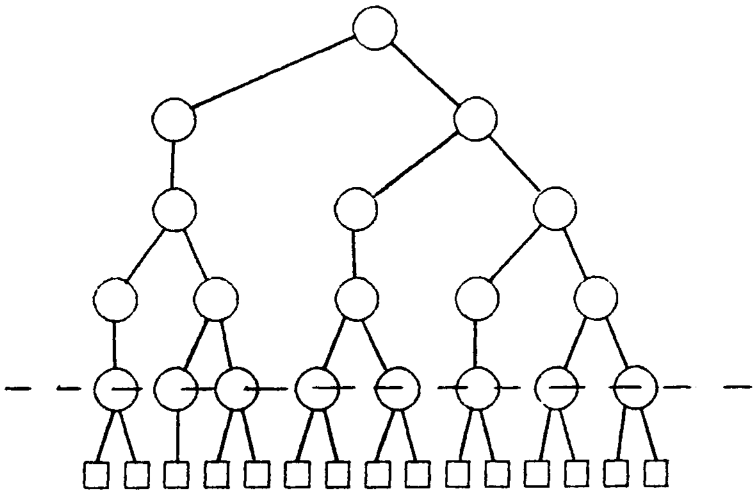
Figures 7–9 give some examples of brother trees which have maximal *NVCOST*; they correspond to different cases in the proof of (5.3.A). The bottommost level at which the number of nodes is Fibonacci is marked by a dotted line.

One question still remains open: How many "non-Fibonacci" levels can a *NVCOST* pessimal brother tree have? In the previous examples we always have $k \leq 2$. This is not by mere accident!



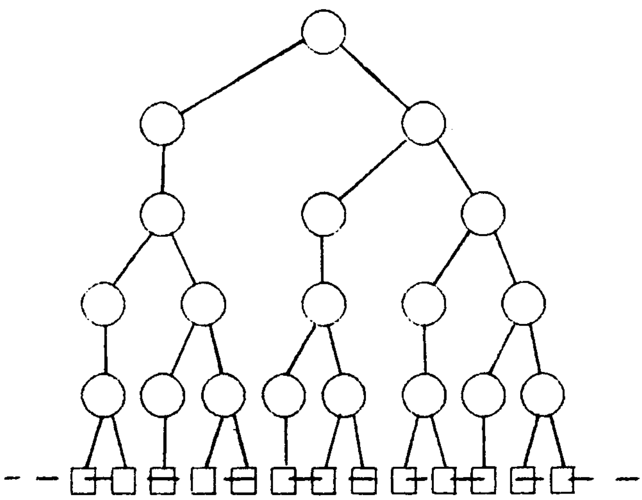
Here $N+1 = 33, h = 6, k = 2, \hat{h} = 3$ because of $8 = f_{3+2} \leq 9 = \left\lceil \frac{33}{2^2} \right\rceil \leq 10 = 2f_{3+1}$. This example shows that we may have $h - k = 4$ in Case 1.

Figure 7.



Here we have $N+1 = 15, h = 5, k = 0, \hat{h} = 4$ due to $13 = f_{4+2} \leq 15 \leq 2f_{4+1} = 16$. With the notations of (4.4), $u + 2 = 3$ holds. Thus $\hat{h} = u + 3$ can occur in Case 2.

Figure 8.



The *NVCOST* pessimal brother tree of size 12 must be a $Fib(5)$ due to (3.3)! We have $h = 5$, $k = 0$, and $\hat{h} = 4$. Here level $\hat{h}+1$ also belongs to the "Fibonacci top".

Figure 9.

(5.3.B) Theorem: *For all integers $N \geq 1$, $\alpha(N) \leq 2$ holds. Thus, a brother tree of size N that has maximal NVCOST is a Fibonacci tree up to at most 3 levels from its bottommost level. More precisely, if $N+1 \in [f_{p+2}, f_{p+3}]$, then*

$$\begin{aligned} \alpha(N) &= 0 \text{ if } N+1 \in [f_{p+2}, 2f_{p+1}], \\ \alpha(N) &= 1 \text{ if } N+1 \in (2f_{p+1}, 4f_p], \\ \alpha(N) &= 2 \text{ if } N+1 \in (4f_p, f_{p+3}). \end{aligned}$$

Moreover, we have $h(T) = p+1$ and $\hat{h} = p - \alpha(N)$.

Proof: First note that each integer $n \geq 2$ lies in a "Fibonacci interval" $[f_{p+2}, f_{p+3})$ for a unique integer $p \geq 0$. We have

$$[f_{p+2}, f_{p+3}) = [f_{p+2}, 2f_{p+1}] \cup (2f_{p+1}, f_{p+3})$$

If n lies in the upper part $(2f_{p+1}, f_{p+3})$ then the Fibonacci interval of $\left\lceil \frac{n}{2} \right\rceil$ is $[f_{p+1}, f_{p+2})$ because $n \leq f_{p+3} - 1$ implies

$$\frac{n}{2} \leq \frac{f_{p+1} + f_{p+2} - 1}{2} \leq \frac{2f_{p+2} - 2}{2} = f_{p+2} - 1$$

Assume $\alpha(N) \geq 1$. By definition of $\alpha(N)$, each number in the sequence

$$N+1, \left\lceil \frac{N+1}{2} \right\rceil, \ldots, \left\lceil \frac{N+1}{2^{\alpha(N)-1}} \right\rceil$$

is included in the upper half of its Fibonacci interval:

$$\begin{aligned} 2f_{p+1} &< N+1 < f_{p+3} \\ 2f_p &< \left\lceil \frac{N+1}{2} \right\rceil < f_{p+2} \\ &\vdots \\ 2f_{p+2-\alpha(N)} &< \left\lceil \frac{N+1}{2^{\alpha(N)-1}} \right\rceil < f_{p+4-\alpha(N)} \end{aligned}$$

The left side of the last inequality implies

$$2^{\alpha(N)} f_{p+2-\alpha(N)} < N+1 < f_{p+3}$$

By a well known identity for Fibonacci numbers,

$$\begin{aligned} f_{p+3} &= f_{p+2-\alpha(N)} f_{\alpha(N)+1} + f_{p+1-\alpha(N)} f_{\alpha(N)} \\ &\leq f_{p+2-\alpha(N)} f_{\alpha(N)+2}. \end{aligned}$$

Together with the above equation this yields

$$2^{\alpha(N)} < f_{\alpha(N)+2}$$

which implies $\alpha(N) \leq 2$. The characterization of α is immediate. In (5.1) we have $\hat{h} = p - \alpha(N)$. With $k = \alpha(N)$ in (5.3.A) we obtain $h(T) = h = \hat{h} + k + 1 = p + 1$. □

Theorems (5.3.A) and (5.3.B) describe completely the structure of brother trees that have maximal *NVCOST*. By (3.2) we already know that a brother tree of size N can have height at most $p+1$ if $N+1 \in [f_{p+2}, f_{p+3})$. Theorem (5.3.B.), in particular, implies that the maximal height equals $p+1$ for all $N+1$ in this interval. Thus, the maximal height of a brother tree is a non-decreasing function of its size.

6 Exact formulae for the minimal and maximal NVCOST of brother trees

In this section we first use the structure theorems of Section 5 to derive precise formulae for the worst case *NVCOST*. Let $\overline{NVCOST(N)}$ denote the maximal node visit cost of all brother trees of size N. Remember that

$$NVCOST(T) = \sum_{i=0}^{h-1}(i+1)\beta_i$$

if T is of height h. From (5.3.A) and (5.3.B) we immediately obtain

$$\overline{NVCOST(N)} = \begin{cases} \sum_{i=0}^{p-1}(i+1)f_i + (p+1)(N+1-f_{p+1}), \\ \qquad \text{if } N+1 \in [f_{p+2}, 2f_{p+1}]. \\ \sum_{i=0}^{p-2}(i+1)f_i + p\left(\left\lceil \frac{N+1}{2} \right\rceil - f_p\right) + (p+1)\left\lfloor \frac{N+1}{2} \right\rfloor, \\ \qquad \text{if } N+1 \in (2f_{p+1}, 4f_p]. \\ \sum_{i=0}^{p-3}(i+1)f_i + (p-1)\left(\left\lceil \frac{N+1}{2^2} \right\rceil - f_{p-1}\right) + \\ p\left\lfloor \frac{\left\lceil \frac{N+1}{2} \right\rceil}{2} \right\rfloor + (p+1)\left\lfloor \frac{N+1}{2} \right\rfloor, \\ \qquad \text{if } N+1 \in (4f_p, f_{p+3}). \end{cases}$$

(6.1) Lemma: $\sum_{i=0}^{p-1}(i+1)f_i = 2 + pf_{p+1} - f_{p+2}$

Proof: By induction on p. □

(6.2) Theorem: *Assume $N+1 \in [f_{p+2}, f_{p+3})$. Then*

$$\overline{NVCOST(N)} = \begin{cases} (p+1)(N+1) - f_{p+3} + 2, \\ \quad \text{if } N+1 \in [f_{p+2}, 2f_{p+1}]. \\ p(N+1) + \left\lfloor \frac{N+1}{2} \right\rfloor - f_{p+2} + 2, \\ \quad \text{if } N+1 \in (2f_{p+1}, 4f_p]. \\ (p-1)(N+1) + \left\lfloor \frac{\left\lceil \frac{N+1}{2} \right\rceil}{2} \right\rfloor + 2\left\lfloor \frac{N+1}{2} \right\rfloor - f_{p+1} + 2, \\ \quad \text{if } N+1 \in (4f_p, f_{p+3}). \end{cases}$$

Proof: Application of (6.1) and the identities

$$\begin{aligned} \left\lfloor \frac{N+1}{2} \right\rfloor + \left\lceil \frac{N+1}{2} \right\rceil &= N+1 \\ \left\lfloor \frac{\left\lceil \frac{N+1}{2} \right\rceil}{2} \right\rfloor + \left\lceil \frac{N+1}{4} \right\rceil + \left\lfloor \frac{N+1}{2} \right\rfloor &= N+1 \end{aligned}$$

□

Disregarding the floor and ceiling functions we can look at $\overline{NVCOST(N)}$ as a function which is continous and piecewise linear with corresponding slopes $p+1$, $p+\frac{1}{2}$, $p+\frac{1}{4}$ inside the Fibonacci interval $[f_{p+2}, 2f_{p+1}]$, but makes a jump of height $\frac{3}{4}f_{p+3} - f_p$ at its right end.

We can compare $\overline{NVCOST(N)}$ with $\underline{NVCOST(N)}$, the minimal node visit cost of all brother trees of size N, which has been computed in [4], (6.3).

(6.3) Theorem: *Assume $N+1 \in (2^{h-1}, 2^h]$. Then*

$$\underline{NVCOST(N)} = \begin{cases} (h-1)(N+1) - 2^{h-2} + 1, \\ \quad \text{if } N+1 \in (2^{h-1}, 3 \cdot 2^{h-2}]. \\ h(N+1) - 2^h + 1, \\ \quad \text{if } N+1 \in (3 \cdot 2^{h-2}, 2^h]. \end{cases}$$

with $h = \lceil \log_2(N+1) \rceil$.

$\underline{NVCOST(N)}$ is a continous and piecewise linear function on $(2^{h-1}, 2^h]$ with slopes $h-1$ and h. It makes a jump of height 2^h at the right end of this interval.

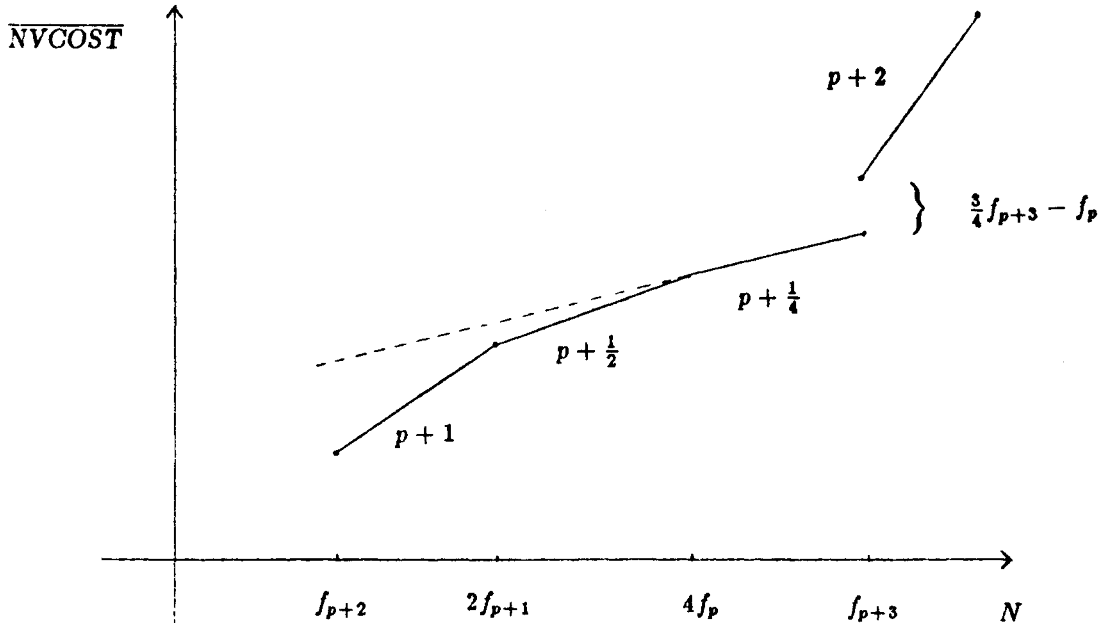


Figure 10: The graph of $\overline{NVCOST(N)}$ in the interval $[f_{p+2}, f_{p+3})$.

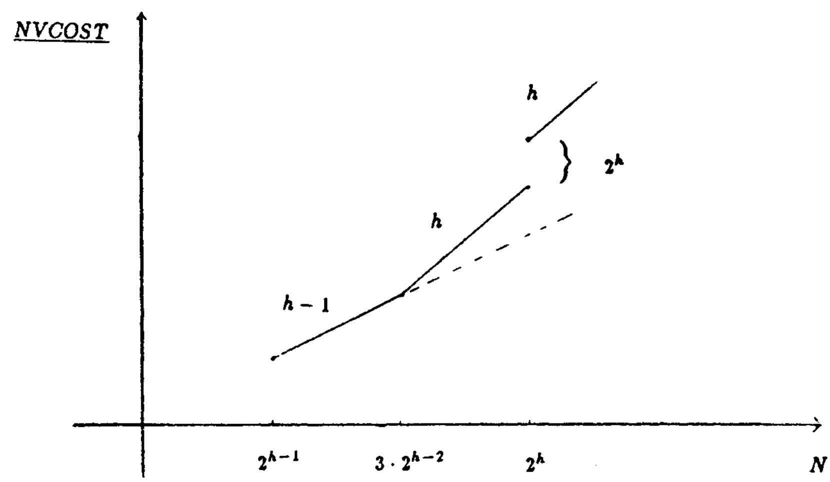


Figure 11: The graph of $\underline{NVCOST(N)}$ in the interval $(2^{h-1}, 2^h]$.

7 Tight bounds for the node visit cost of brother trees

First we want to derive a more useful, tight upper bound for $\overline{NVCOST(N)}$. In Figure 10 the line with slope $p + \frac{1}{4}$ majorizes the graph of $NVCOST$ in the interval $[f_{p+2}, f_{p+3})$. Thus,

$$\overline{NVCOST(N)} \leq (p + \frac{1}{4})(N+1) - f_{p+1} + 2$$

for all $N + 1 \in [f_{p+2}, f_{p+3})$. Here we use

$$\left\lfloor \frac{\left\lceil \frac{N+1}{2} \right\rceil}{2} \right\rfloor + 2 \left\lfloor \frac{N+1}{2} \right\rfloor \leq \frac{5}{4}(N+1),$$

where equality holds if and only if $N + 1$ is a multiple of 4. In order to get rid of the term f_{p+1} remember that

$$f_n = \frac{1}{\sqrt{5}}(\alpha^{n+1} - \bar{\alpha}^{n+1}), \quad n \geq 0$$

holds for the Fibonacci numbers; here $\alpha = \frac{1+\sqrt{5}}{2} \simeq 1.618$ is the positive root of $X^2 - X - 1 = 0$, and $\bar{\alpha} = \frac{1-\sqrt{5}}{2} \simeq -0.618$ is the negative one. Hence, using $\alpha\bar{\alpha} = -1$,

$$\begin{aligned}
\frac{f_{p+1}}{f_{p+3}-1} - \left(\frac{1}{\alpha^2} + \frac{1}{\alpha^2}\frac{1}{f_{p+3}-1}\right) &= \frac{\bar{\alpha}^{p+4} - \alpha^2\bar{\alpha}^{p+2}}{\alpha^{p+6} - \alpha^2\bar{\alpha}^{p+4} - \sqrt{5}\alpha^2} \\
&= \frac{(-1)^{p+1}\sqrt{5}}{\alpha^{2p+8} + (-1)^{p+1} - \sqrt{5}\alpha^{p+4}} \\
&= O\left(\frac{1}{(N+1)^2}\right)
\end{aligned}$$

for $N + 1 < f_{p+3} = O(\alpha^p)$. Therefore,

$$\begin{aligned}
f_{p+1} &= \left(\frac{1}{\alpha^2} + \frac{1}{\alpha^2}\frac{1}{f_{p+3}-1} + O\left(\frac{1}{(N+1)^2}\right)\right) \cdot \frac{f_{p+3}-1}{N+1} \cdot (N+1) \\
&\geq \frac{1}{\alpha^2}(N+1) + \frac{1}{\alpha^2} + O\left(\frac{1}{N+1}\right) \quad \text{for} \quad \frac{f_{p+3}-1}{N+1} \geq 1
\end{aligned}$$

Because $2 - \frac{1}{\alpha^2} = \alpha$, this yields

(7.1) Lemma:

$$\overline{NVCOST(N)} \leq (p + \frac{1}{4} - \frac{1}{\alpha^2})(N+1) + \alpha + O\left(\frac{1}{N+1}\right)$$

for $N+1 \in [f_{p+2}, f_{p+3})$. Equality holds for all $N+1 = f_{p+3} - 1 \equiv 0$ mod 4.

For the minimal node visit cost, we see from Figure 11 that

$$\underline{NVCOST(N)} \geq (h-1)(N+1) - 2^{h-2} + 1$$

holds for $N+1 \in (2^{h-1}, 2^h]$. Since $2^{h-2} + \frac{1}{2} \leq \frac{1}{2}(N+1)$ we obtain

$$\underline{NVCOST(N)} \geq \left(h - \frac{3}{2}\right)(N+1) + \frac{3}{2}$$

Equality holds for $N+1 = 2^{h-1}+1$. Summarizing we get

(7.2) Theorem: *If T is a brother tree of size N then*

$$\begin{aligned}(h - \frac{3}{2})(N+1) + \frac{3}{2} &\leq NVCOST(T) \\ &\leq (p-c)(N+1) + \alpha + O\left(\frac{1}{N+1}\right)\end{aligned}$$

holds and both bounds are tight. Here

$$\begin{aligned} h &= \lceil \log_2(N+1) \rceil \\ p &= \left\lceil \frac{\log_2(N+\frac{3}{2}) + \log_2(\sqrt{5})}{\log_2(\alpha)} - 4 \right\rceil \\ &\simeq \lceil 1.44 \cdot \log_2(N + \frac{3}{2}) - 2.328 \rceil \\ \alpha &= \frac{1+\sqrt{5}}{2} \simeq 1.618 \\ c &= \frac{5 - 2\sqrt{5}}{4} \simeq 0.132 \end{aligned}$$

Proof: We have $N+1 \in [f_{p+2}, f_{p+3})$ if and only if f_{p+3} is the smallest Fibonacci number greater than $N+1$, or, equivalently, p is the smallest integer such that

$$N + \frac{3}{2} \leq \frac{1}{\sqrt{5}}\alpha^{p+4}$$

For, $\left|\frac{1}{\sqrt{5}}\bar{\alpha}^{p+4}\right| < \frac{1}{2}$ □

(7.3) Corollary: *Up to an* $O\left(\frac{\log_2 N}{N}\right)$ *error, we have*

$$\lceil \log_2(N+1) \rceil - \frac{3}{2} \leq \frac{NVCOST(T)}{N} \leq \lceil 1.44 \log_2(N + \frac{3}{2}) - 0.328 \rceil - 2.132$$

for the average number of node visits per access in a brother tree of size N.

By Theorem (5.3.B) and (2.3), the maximal height $\bar{h}(N)$ of a brother tree of size N is

$$\bar{h}(N) = p + 1 \simeq \lceil 1.44 \log_2(N + \frac{3}{2}) - 0.328 \rceil - 1$$

Hence, the upper bound in (7.3) can be rewritten as

(*i*) $$\frac{\overline{NVCOST(N)}}{N} \leq \bar{h}(N) - 1 - c + O\left(\frac{\log N}{N}\right)$$

On the other hand, the formula before Corollary (2.4) tells us that

(*ii*) $$\overline{NVCOST(N)} = \bar{h}(N)(N+1) - N - \underline{\omega}_N,$$

where $\underline{\omega}_N$ denotes the minimum number of unary nodes in a tree of size N that is of maximal height.

(7.4) Corollary: *If a brother tree T is sufficiently large and of maximal height, then at least* 11.6% *of its internal nodes are unary.*

Proof: Combining (i) with (ii) yields

$$\frac{\underline{\omega}_N}{N} \geq c + O\left(\frac{\log N}{N}\right)$$

Thus

$$\frac{\underline{\omega}_N}{\underline{\omega}_N + N} \geq \frac{c}{c+1} + O\left(\frac{\log N}{N}\right)$$

with

$$\frac{c}{c+1} = \frac{25 - 8\sqrt{5}}{61} \simeq 0.1166$$

□

This minimum value is obtained if $N+1$ is a multiple of 4, is a predecessor of a Fibonacci number, and if T has maximum $NVCOST$ (see (7.1)). Note that the maximum percentage of unary nodes is approximately $\frac{100}{\alpha^2} = 38.2\%$ and this is obtained in Fibonacci trees.

8 Concluding Remarks

In this paper we have continued the investigation of the node visit cost, $NVCOST(T)$, of a brother tree T, an important time cost measure for this balanced tree scheme, which has been introduced and studied in [4]. We first characterized the worst case structures, that is, the structure of those brother trees whose $NVCOST$ is a maximum among all brother trees of size N. From this structure theorem we derived both an exact formula and a tight upper bound for the maximal node visit cost. The latter, together with a lower bound for $NVCOST$ that has been derived in [4], determines the range of this cost precisely. Thereby, one of the open problems posed in [4] has been solved. This solution was not trivial, nevertheless, we think it is worth trying to get similar results for other cost measures (for example, comparison cost) and other balancing schemes (for example, AVL trees).

9 Acknowledgement

The first author wants to thank his wife Anne Brüggemann-Klein for introducing him to TEX and for her TEXnical help.

References

[1] G.M. Adel'son-Vel'skii and Y.M. Landis. An algorithm for the organization of information. *Doklady Akademi Nauk*, 146:263–266, 1962.

[2] K. Mehlhorn. *Data Structures and Algorithms 1: Sorting and Searching.* Springer-Verlag, New York, 1984.

[3] J. Nievergelt and E.M. Reingold. Binary search trees of bounded balance. *SIAM Journal on Computing*, 2:33–43, 1973.

[4] Th. Ottmann, D.St. Parker, A.L. Rosenberg, H.-W. Six, and D. Wood. Minimal-cost brother trees. *SIAM Journal on Computing*, 13:197–217, 1984.

[5] Th. Ottmann and H.-W. Six. Eine neue Klasse von ausgeglichen Binärbäumen. *Angewandte Informatik*, 9:395–400, 1976.

[6] Th. Ottmann, H.-W. Six, and D. Wood. On the correspondence between AVL trees and brother trees. *Computing*, 23:43–54, 1979.

[7] Th. Ottmann, H.-W. Six, and D. Wood. Right brother trees. *Communications of the ACM*, 21:769–776, 1981.

[8] Th. Ottmann and D. Wood. 1-2 brother trees or AVL trees revisited. *Computer Journal*, 23:248–255, 1981.