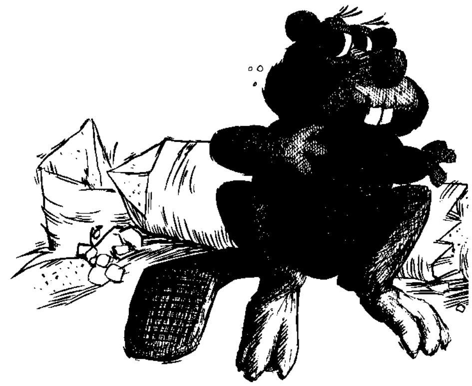
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



A Tight Upper Bound for the Path Length of AVL Trees

Rolf Klein
Derick Wood

Data Structuring Group

Research Report
CS-87-17

March, 1987

A Tight Upper Bound for the Path Length of AVL Trees *

Rolf Klein † Derick Wood‡

March 11, 1987

Abstract

We prove that the internal path length of an AVL tree of size N is bounded from above by

$$1.4404N(\log_2 N - \log_2 \log_2 N) + O(N)$$

and show that this bound is achieved by an infinite family of AVL trees. But AVL trees of maximal height do not have maximal path length. These results carry over to the comparison cost of brother trees.

Key words: AVL trees, brother trees, path length, comparison cost, node visit cost.

1 Introduction

The cost of a search operation in a tree corresponds to the length of the path from the root to the node that contains the desired information. Almost 25 years ago, Adel'son-Vel'skii and Landis [1] introduced AVL trees, the first class of what came to be known as balanced trees. They satisfy the basic property that their height is logarithmic in their size. Although the worst case height of AVL trees is well known to be $1.4404 \log_2 N$ (see [1,7]), the worst case internal path length (or IPL) has been an open problem. In this paper we provide the first tight upper bound on the internal path length of AVL trees. Moreover, we demonstrate why AVL trees of maximal height

*This work was partially supported by a Natural Sciences and Engineering Research Council of Canada Grant A-5692. It was done during the first author's stay with the Data Structuring Group in Waterloo in 1986/1987.

†Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, Postfach 6980, 7500 Karlsruhe, West Germany

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

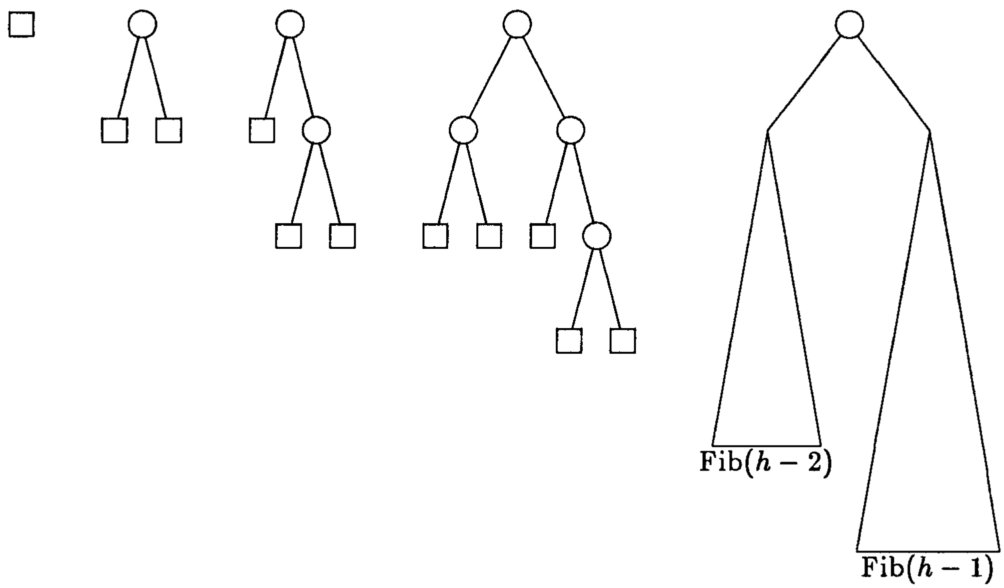


Figure 1: Fibonacci trees

cannot have maximal internal path length. This illustrates that the principle of the AVL tree scheme, namely, keeping access paths short by balancing the heights of subtrees, does not guarantee good worst case behaviour of the internal path length.

Since each path's length is bounded by the height there is an obvious upper bound for the internal path length of an AVL tree T, namely

$$IPL(T) \leq 1.4404 N \log_2 N$$

But can this rough bound really be achieved? We shall show that it almost can (up to a lower order term), but that the AVL trees of maximal path length do not look like one might expect, because they are not of maximal height.

If $N+1$ is a Fibonacci number F_{h+2}, then the AVL tree of maximal height is uniquely determined; it is the *Fibonacci tree* Fib(h). In Figure 1[1] we display the AVL trees $Fib(0), Fib(1), Fib(2), Fib(3)$, and $Fib(h)$. But, despite their being of maximal height Fibonacci trees do not have maximal path length! For example, consider $Fib(6)$ of size 20, height 6, and internal path length 76. The tree T_{20} of Figure 2 has the same size but height 5 and IPL 77, as was already noted by Knuth [7], Exercise 4, p. 470. Since $Fib(7)$ also fails to have maximal path length, this implies that $Fib(h)$ is not IPL

[1] This and the following figures were drawn using TreeTEX[2]

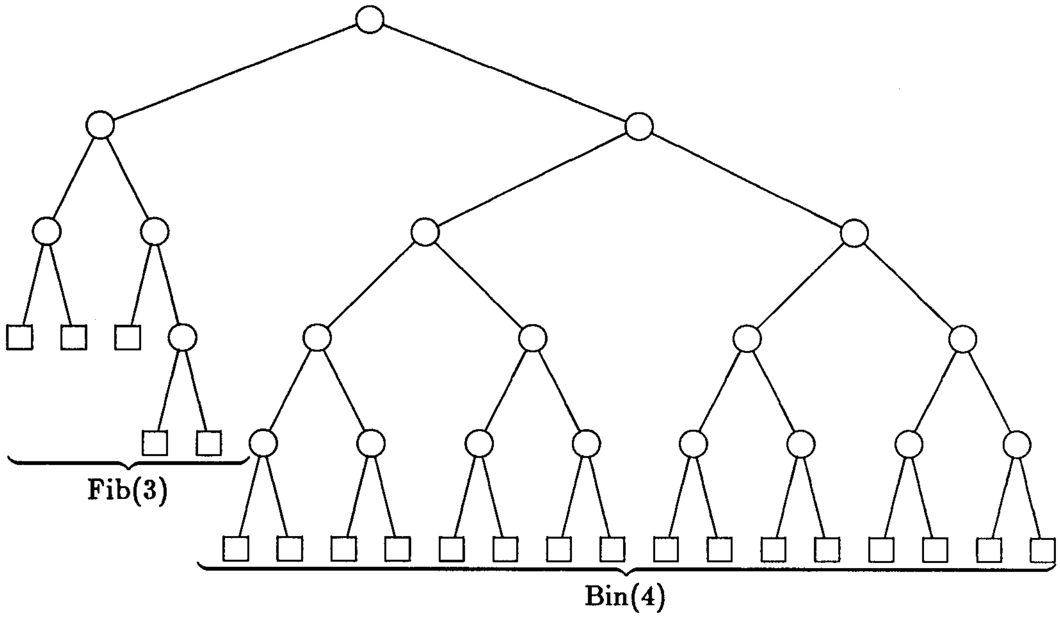


Figure 2: A counterexample tree

pessimal, for all $h \geq 6$. In fact, Fibonacci trees have an internal path length

$$IPL(Fib(h)) = 1.0422N\log_2 N + O(N)$$

(see Lemma 2.3, B), which is not pessimal but rather quite close to the optimum!

Independently, Gonnet [4] also made this observation. He presented a family $\mathcal{G}$ of AVL trees some of which have an IPL of

$$1.2557N\log_2 N + O(N)$$

The trees in this family $\mathcal{G}$, like the tree of Figure 2, have only two kinds of subtrees: complete binary trees and Fibonacci trees.

In the next section we prove that the internal path length of a binary tree is bounded from above by

$$1.4404N\log_2 N - 1.4404N\log_2\log_2 N + O(N)$$

Then, we show in section 3 that this bound can be achieved by AVL trees that are also members of the family $\mathcal{G}$ — but worse ones. In order to establish our upper bound we first prove that

$$IPL(T) \leq h(N+1) - \frac{1}{5}h\phi^{h+1} + 1$$

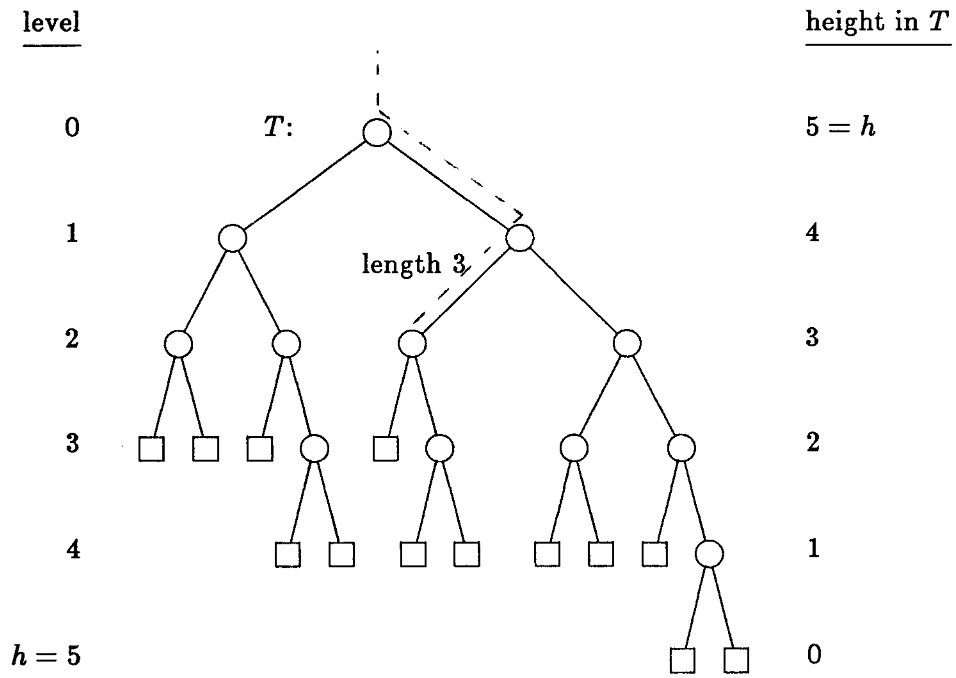


Figure 3: Notations

holds for each AVL tree of size N and height h, where $\phi = (1+\sqrt{5})/2 = 1.618\ldots$. This estimation shows why AVL trees of maximal height cannot have maximal path length. For, if h is increased beyond a certain threshold then the exponential growth rate of the negative term will outrun the linear positive term and, therefore, the value will decrease.

In the last section we show that our results on the internal path length of AVL trees carry over to the comparison cost of brother trees. This cost measure together with the node visit cost measure, captures the time complexity of this tree scheme; see [5].

2 An upper bound for the internal path length

Let T be an extended binary tree. We count the *level* of nodes starting with level 0 at the root. The *height* h of T is its maximum level number and its *size* is the number N of internal nodes. A node at level i is said to be at *height* $h - i$ *with respect to* T. The *access path* to a node at level i is of *length* $i+1$ (see Figure 3). Furthermore, *weight*$(T) = \text{size}(T) + 1$ denotes the number of external nodes of T.

The *internal path length* of a binary tree T is defined as

$$IPL(T) = \sum_{p \text{ binary}} length(path(p))$$

Note that the *external path length* EPL of a tree of size N, that is, the sum of the lengths of all paths from the root to the external nodes, is related to IPL by the formula

$$IPL(T) + 2N + 1 = EPL(T)$$

For, each internal node p of T with rooting subtree T_p is counted size(T_p) times in computing IPL, but weight(T_p) times in the computation of EPL. Furthermore, each of the $N+1$ external nodes contributes to EPL. The difference being $O(N)$, all our asymptotic results hold for EPL as well as for IPL. Finally, let us recall that an *AVL tree* is an extended binary tree in which the heights of the two subtrees of each internal node differ by at most one.

First we derive a formula for IPL that corresponds to a view from the frontier of the tree.

Lemma 2.1 *Let T be an AVL tree of size N and height h, and h_i be the height of the i-th external node in T. Then,*

$$IPL(T) = (h-1)(N+1) + 1 - \sum_{i=1}^{N+1} h_i$$

Proof: We have

$$IPL(T) + 2N + 1 = EPL(T) = (h+1)(N+1) - \sum_{i=1}^{N+1} h_i$$

see Figure 4. □

We denote $\sum_{i=1}^{N+1} h_i$ by $U(T)$ in the following; $U(T)$ is the "area" below the external nodes of height ≥ 1 in T; see Figure 4. The formula in Lemma 2.1 does not hold only for AVL trees but also for arbitrary binary trees. It can be used to establish an interesting connection between the path length of a binary tree and the ratio of the arithmetic and the geometric mean of certain integers; see [6].

To derive an upper bound for IPL we need to establish a lower bound for $U(T)$, according to the above Lemma. If T is a complete binary tree then $U(T) = 0$. Intuitively, any height exceeding $\log_2 N$ must be paid for by a "skew" frontier and thereby, by a non-zero area $U(T)$. In the following

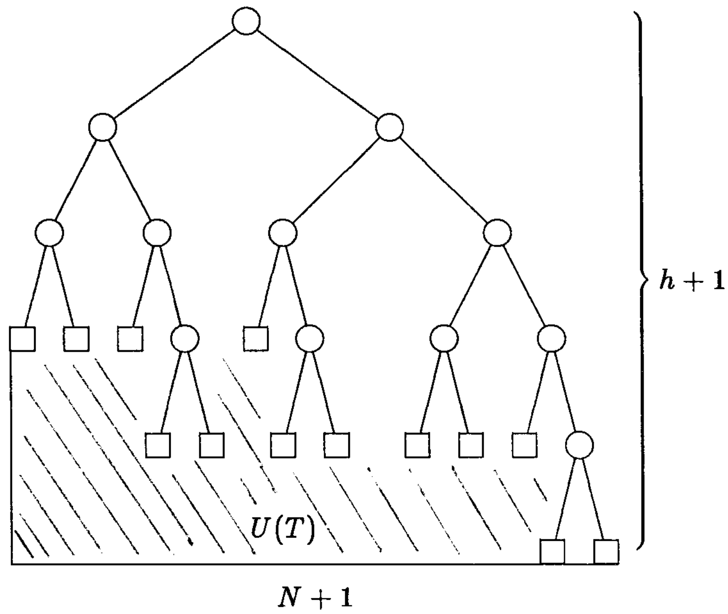


Figure 4: The area $U(T)$

Lemma we compute $U(T)$ for those AVL trees which are as skew as possible, i.e., the Fibonacci trees. Recall that the Fibonacci tree $Fib(h)$ is of height h and of weight F_{h+2}, where $F_0 = 0, F_1 = 1, F_{h+2} = F_h + F_{h+1}$ denotes the sequence of Fibonacci numbers.

Lemma 2.2

$$U(Fib(h)) = \frac{1}{5}(hF_{h+2} + (h-3)F_h)$$

Proof: By induction on h. If $h = 0$ or $h = 1$, then both sides are equal to zero. Assume $h \geq 2$. Due to the recursive definition of Fibonacci trees (see Figure 1) we have

$$U(Fib(h)) = U(Fib(h-2)) + F_h + U(Fib(h-1))$$

because the subtree $Fib(h-2)$ is lifted by one level. By the induction hypothesis, this yields

$$\begin{aligned} U(Fib(h)) &= \frac{1}{5}((h-2)F_h + (h-5)F_{h-2} + 5F_h + (h-1)F_{h+1} + (h-4)F_{h-1}) \\ &= \frac{1}{5}(hF_{h+2} + (h-3)F_h) \end{aligned}$$

□

We derive an asymptotic formula for $U(Fib(h))$.

Lemma 2.3
A. $U(Fib(h)) \geq \frac{1}{5}h\phi^{h+1} - \phi^h$
B. $IPL(Fib(h)) = 1.0422928\ldots N\log_2 N + O(N)$

Proof: It is well known that

$$F_h = \frac{1}{\sqrt{5}}(\phi^h - \hat{\phi}^h)$$

holds, where $\phi = (1+\sqrt{5})/2 = 1.618\ldots$ is the positive root of $X^2 - X - 1$, and $\hat{\phi} = 1 - \phi = -0.618\ldots$ is the negative one. By Lemma 2.2,

$$\begin{aligned} U(Fib(h)) &\geq \frac{1}{5}\frac{1}{\sqrt{5}}h(\phi^{h+2} + \phi^h) - \phi^h \\ &\geq \frac{1}{5}h\phi^{h+1} - \phi^h \end{aligned}$$

because $\phi + \phi^{-1} = 2\phi - 1 = \sqrt{5}$. This proves assertion A. Since $\phi^{h+1} = \sqrt{5}\phi^{-1}F_{h+2} + o(1)$ Lemma 2.1 yields

$$\begin{aligned} IPL(Fib(h)) &= hF_{h+2} - \frac{1}{5}h\sqrt{5}\phi^{-1}F_{h+2} + O(F_{h+2}) \\ &= \frac{2+\phi}{5}hF_{h+2} + O(F_{h+2}) \end{aligned}$$

Because $h = \log_\phi F_{h+2} + O(1)$

$$\begin{aligned} IPL(Fib(h)) &= \frac{2+\phi}{5}\log_\phi 2N\log_2 N + O(N) \\ &= 1.042298\ldots N\log_2 N + O(N) \end{aligned}$$

follows, where $N = F_{h+2} = \text{weight}(Fib(h))$. □

Now the crucial point in deriving a lower bound for $U(T)$ is to consider a linear combination of U and the weight, rather than U alone.

Definition 2.1 *For an AVL tree T and integers $a, h \geq 0$, let*

$$U(T,a) = U(T) + a \cdot weight(T)$$

and

$$U(h,a) = \min(\{U(T,a); T \text{ is an AVL tree of height } h\})$$

Figure 5 shows the graphical representation of $U(T,a)$.

If $a = 0$ then $U(h,a) = 0$. The following Lemma shows that for each integer $a \geq 1$ the Fibonacci tree has a minimum area $U(T,a)$ among all AVL trees T of height h.

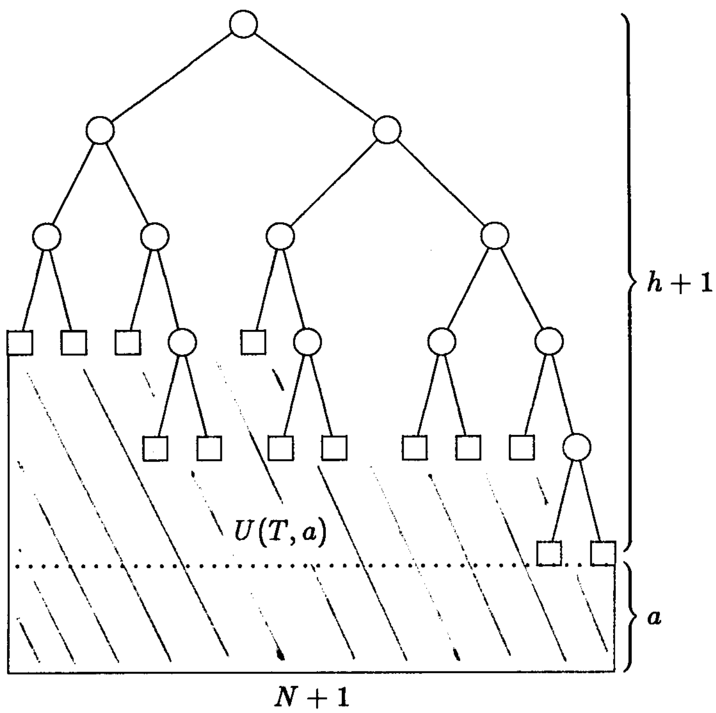


Figure 5: The area $U(T,a)$

Lemma 2.4 *Let $h \geq 0$ and $a \geq 1$; then* $U(h,a) = U(Fib(h),a)$.

Proof: By induction on h. The cases $h = 0$ and $h = 1$ are trivial because $Fib(0)$ and $Fib(1)$ are the only AVL trees of their height. Assume $h \geq 2$ and $U(h,a) = U(T,a)$, for an AVL tree T of height h. Two cases arise, according to the structure of T; see Figure 6.

Case 1. Both subtrees are of height $h-1$. We have $U(h,a) = U(T,a) = U(T_l,a) + U(T_r,a)$, hence $U(T_l,a) = U(h-1,a) = U(T_r,a)$, by the minimality of T. By the induction hypothesis,

$$U(T,a) = 2U(Fib(h-1),a)$$

Case 2. T_l is of height $h-2$, T_r of height $h-1$. Here $U(h,a) = U(T,a) = U(T_l,a+1) + U(T_r,a)$, hence $U(T_l,a+1) = U(h-2,a+1)$ and $U(T_r,a) = U(h-1,a)$. By the induction hypothesis,

$$U(T,a) = U(Fib(h-2),a+1) + U(Fib(h-1),a)$$

We claim that Case 2 applies because the corresponding value of $U(T,a)$ is the minimum of both. We have to show that

$$U(Fib(h-2)) + (a+1)F_h \leq U(Fib(h-1)) + aF_{h+1}$$

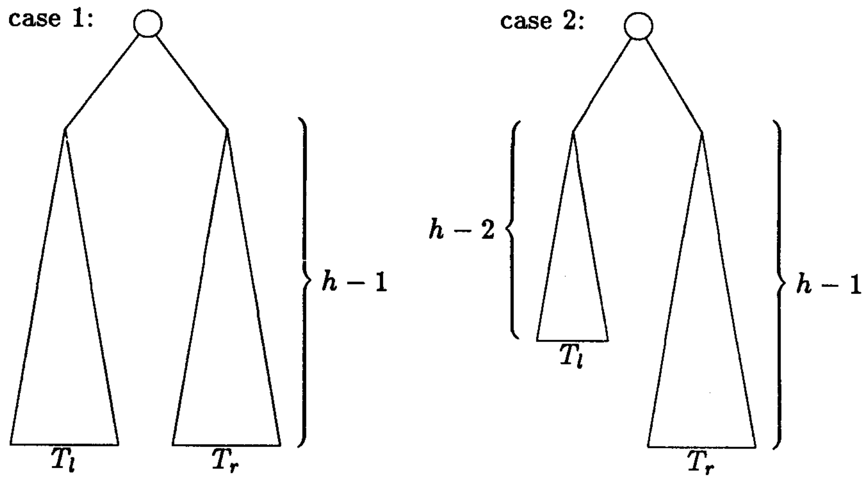


Figure 6: The possible structure of T

or, equivalently

$$F_h \leq U(Fib(h-1)) - U(Fib(h-2)) + aF_{h-1}$$

holds. This is true if $h = 2$. If $h \geq 3$ then

$$U(Fib(h-1)) - U(Fib(h-2)) = U(Fib(h-3)) + F_{h-1} \geq F_{h-1}$$

(see the proof of Lemma 2.2), and the assertion follows because, for $a \geq 1$, $F_h \leq (a+1)F_{h-1}$ holds. Therefore,

$$\begin{aligned} U(h,a) &= U(T,a) \\ &= U(Fib(h-2), a+1) + U(Fib(h-1), a) \\ &= U(Fib(h), a) \end{aligned}$$

□

This theorem provides us with the appropriate tool.

Theorem 2.5 *Let T be an AVL tree of height h and size N. Then*

$$IPL(T) \leq h(N+1) - \frac{1}{5}h\phi^{h+1} + 1$$

Proof: An application of Lemma 2.5 with $a = 1$ yields

$$\begin{aligned} U(T) + (N+1) &\geq U(h,1) = U(Fib(h)) + F_{h+2} \\ &\geq \frac{1}{5}h\phi^{h+1} - \phi^h + F_{h+2} \\ &\geq \frac{1}{5}h\phi^{h+1} \end{aligned}$$

according to Lemma 2.3, A. Now we obtain

$$IPL(T) \leq h(N+1) - \frac{1}{5}h\phi^{h+1} + 1$$

by Lemma 2.1. $\square$

Theorem 2.6 *Let T be an AVL tree of size N. Then*

$$IPL(T) \leq 1.4404\ldots N(\log_2 N - \log_2\log_2 N) + O(N)$$

Proof: Let $C = \phi/5$. The function

$$f(X) = X(N+1) - CX\phi^X$$

takes its maximum at the zero of its first derivative

$$\frac{df}{dX} = (N+1) - C(\phi^X + X\phi^X \ln\phi),$$

that is, at x satisfying the equation

$$\phi^x(1 + x\ln\phi) = \frac{1}{C}(N+1)$$

This yields

$$x\phi^x = \Theta(N) \tag{1}$$

and

$$x + \log_\phi x = \log_\phi N + O(1) \tag{2}$$

After adding $\log_\phi\log_\phi N - \log_\phi x$ to either side in (2) we get

$$\begin{aligned} x + \log_\phi\log_\phi N &= \log_\phi N + \log_\phi\left(\frac{\log_\phi N}{x}\right) + O(1) \\ &= \log_\phi N + \log_\phi\left(1 + \frac{\log_\phi x + O(1)}{x}\right) + O(1) \\ &= \log_\phi N + O(1) \end{aligned}$$

using (2) again. Therefore, f takes its maximum at

$$x = \log_\phi N - \log_\phi\log_\phi N + O(1) \tag{3}$$

Now Theorem 2.6 yields

$$\begin{aligned} IPL(T) &\leq f(h) + O(N) \leq f(x) + O(N) \\ &\leq N(\log_\phi N - \log_\phi\log_\phi N) + O(N) \\ &\leq 1.4404\ldots N(\log_2 N - \log_2\log_2 N) + O(N) \end{aligned}$$

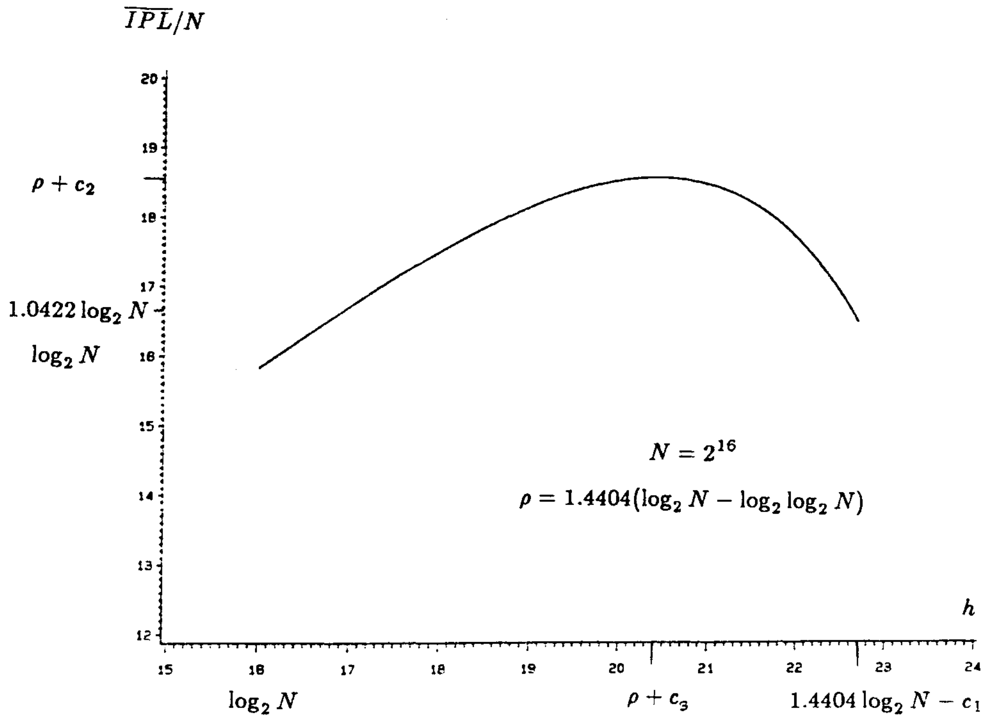


Figure 7: The upper bound

using (3), (1), and $\log_\phi Y = 1.4404 \ldots \log_2 Y$. $\square$

In the next section we show that the maximum value determined above can in fact be achieved by AVL trees whose average internal path length is equal to their height.

Now the reason why AVL trees of maximal height can't have a maximal internal path length has become clear: If h is increased beyond $\log_\phi N - \log_\phi \log_\phi N$ then the upper bound in Theorem 2.4 begins to decrease. Eventually, when h takes its maximal value $\log_\phi N + \log_\phi \sqrt{5} - 2$, the value of the bound is down to

$$
\begin{aligned}
h\frac{1}{\sqrt{5}}\phi^{h+2} - \frac{1}{5}h\phi^{h+1} + O(1) &= \frac{2+\phi}{5}h\frac{1}{\sqrt{5}}\phi^{h+2} + O(1) \\
&= IPL(Fib(h)) \\
&= 1.0422N\log_2 N + O(N)
\end{aligned}
$$

see Lemma 2.3, B. Figure 7 displays the graph of $\overline{\text{IPL}}/N$, the upper bound for the average path length, for a fixed size N.

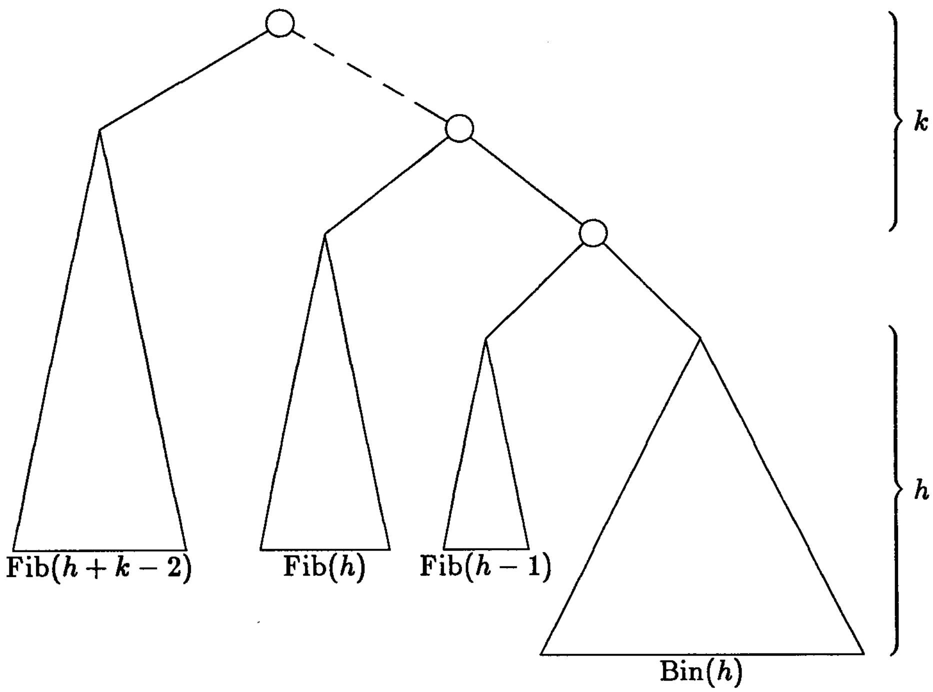


Figure 8: The tree $G(k,h)$

3 Asymptotically IPL pessimal AVL trees

In order to prove that the upper bound obtained in Theorem 2.7 is in fact achievable we consider the family of AVL trees

$$\mathcal{G} = \{G(k,h); k,h \text{ are integers} \geq 0\},$$

where $G(k,h)$ is the AVL tree obtained by replacing the "lowest" subtree $Fib(h)$ of a Fibonacci tree $Fib(h+k)$ by a complete binary tree $\mathrm{Bin}(h)$, see Figure 8.

Gonnet [4] has shown that for $k \sim (\log_\phi 2 - 1)h$

$$\begin{aligned} IPL(G(k,h)) &= \left(1 - \frac{\sqrt{5}}{15}\right) N \log_\phi N + O(N) \\ &= 1.225695\ldots N \log_2 N + O(N) \end{aligned}$$

holds. We show

Theorem 3.1 *Let* $k = (\log_\phi 2 - 1)h - \log_\phi h + \epsilon$, *where* $|\, \epsilon \,| \leq \frac{1}{2}$. *Then*

$$IPL(G(k,h)) = N \log_\phi N - N \log_\phi \log_\phi N + O(N)$$

Proof: Since $\log_\phi 2 - 1 \sim 0.4$ we have $k < h$. Now

$$\begin{aligned} height(G(k,h)) &= k + h \\ &= h\log_\phi 2 - \log_\phi h + \epsilon \\ &= \log_\phi(\frac{2^h}{h}) + \epsilon \end{aligned} \tag{4}$$

and

$$\begin{aligned} N+1 &= F_{h+k+2} - F_{h+2} + 2^h \\ &= g\alpha(k)\phi^{h+k} + 2^h + O(1) \\ &= (\frac{g\alpha(k)\phi^\epsilon}{h} + 1)2^h + O(1) \end{aligned} \tag{5}$$

where $g = \frac{1}{\sqrt{5}}\phi^2$ and $\alpha(k) = 1 - \frac{1}{\phi^k}$, since

$$\phi^{h+k} = \frac{2^h}{h}\phi^\epsilon \tag{6}$$

from (4). Now by Lemma 2.3, A, for some constant $d > 0$ we have

$$\begin{aligned} U(G(k,h)) &\leq U(Fib(k+h)) \\ &\leq d(h+k)\phi^{h+k} \\ &\leq d\phi^\epsilon(1+\frac{k}{h})2^h < 2d\phi^\epsilon 2^h \\ &= O(N) \end{aligned}$$

by (6) and because $k < h$, and by (5). We also have, from (5) and (6) that

$$N+1 = (g\alpha(k) + h\phi^{-\epsilon})\phi^{h+k} + O(1) \tag{7}$$

hence, taking logs we obtain

$$h + k + \log_\phi(g\alpha(k) + h\phi^{-\epsilon}) = \log_\phi N + O(1) \tag{8}$$

Now

$$\begin{aligned} \log_\phi(g\alpha(k) + h\phi^{-\epsilon}) &= \log_\phi h + O(1) \\ &= \log_\phi \log_\phi N + O(1) \end{aligned}$$

because from (5) we have

$$\begin{aligned} h &= \log_2 N + O(1) \\ &= \log_2 \phi \log_\phi N + O(1) \end{aligned}$$

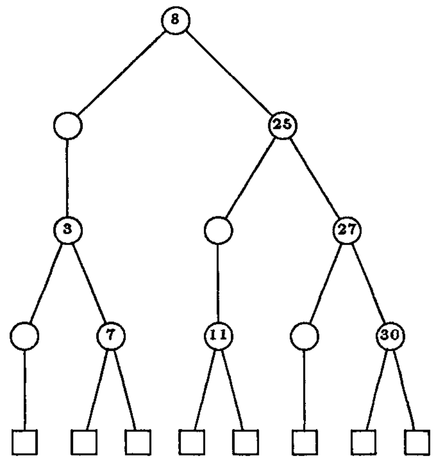


Figure 9: A brother search tree

Summarizing, we have

$$h + k = \log_\phi N - \log_\phi \log_\phi N + O(1)$$

and Lemma 2.1 yields

$$\begin{aligned} IPL(G(k,h)) &= N height(G(k,h)) - U(G(k,h)) + O(N) \\ &= N(\log_\phi N - \log_\phi \log_\phi N) + O(N) \end{aligned}$$

□

This shows that the upper bound in Theorem 2.7 is tight. The asymptotically IPL pessimal AVL trees constructed here have an average path length that differs from the height by an additive constant only!

4 The comparison cost of brother trees

A *brother tree* is a rooted, directed tree each of whose internal nodes has either one or two sons. Each unary node must have a binary brother. All external nodes are at the same level. We associate one key to each internal binary node while the internal unary nodes and the external nodes remain empty. The keys are stored in inorder. This results in the class of *brother search trees* (also called 1-2 *brother trees* in the literature); see Figure 9 and [9].

During a search in a brother search tree, both the unary and the binary nodes on the search path must be *visited* but *key comparisons* only occur at the binary nodes on the path. Therefore, the time complexity of a brother tree T has two constituent parts, *node-visit cost* or NVCOST and *comparison cost* or CCOST, where

$$NVCOST(T) = \sum_{p \text{ internal binary}} \text{number of nodes on path}(p)$$

and

$$CCOST(T) = \sum_{p \text{ internal binary}} \text{number of binary nodes on path}(p).$$

The space cost SCOST of a brother tree is the number of internal binary and unary nodes.

In [8] the structure of those brother trees that are optimal with respect to one of these cost measures has been determined as well as how to construct them in linear time. From these structural results, tight lower bounds for the cost measures can be derived. In [5] a characterization of the structure of all NVCOST-pessimal brother trees has been given, which led to a tight upper bound for the node visit cost.

Although we do not have a structural result on CCOST pessimal brother trees Theorem 2.7 and Theorem 3.1 provide us with a tight upper bound for the comparison cost. For, the following correspondence holds between brother trees and AVL trees (see [10] and [11] for details). The contraction of a brother tree performed by removing its unary nodes results in an AVL tree, and by this operation each AVL tree is obtained exactly once. Clearly, CCOST corresponds to IPL. Therefore

$$CCOST(T) \leq 1.4404\ldots N(\log_2 N - \log_2\log_2 N) + O(N)$$

holds for each brother tree T of size N, and this upper bound is tight.

5 Concluding Remarks

We have shown that

$$1.4404N(\log_2 N - \log_2\log_2 N) + O(N)$$

is a tight upper bound for the internal path length of AVL trees and for the comparison cost of brother trees. But it remains an open problem to characterize the structure of those AVL trees that have maximal internal path length, for any given size N. In Foster [3] the attempt was made to construct IPL pessimal AVL trees of given size and given height, but the algorithm suggested there was incorrect, see Knuth [7], p. 675.

We have also shown that for a fixed size of the tree the maximal internal path length is not increasing with the height but rather increasing first and then decreasing, after taking its maximum as the behavior of the upper bound in Theorem 2.6 suggests. The maximum is achieved by AVL trees whose average internal path length is equal to their height, up to an additive constant. This illustrates that the AVL tree principle, to balance the height of subtrees does not guarantee a short internal path. In fact, Gonnet [4] shows that balancing the internal path length directly results in an upper bound for IPL of only

$$1.05155\ldots N \log_2 N + O(N)$$

which is only 5 percent worse than optimal, whereas the path length of AVL trees can be 44 percent worse than optimal. However, in these trees insertions and deletions can cause linear cost in the (very unlikely) worst case. Therefore, the question arises: is there a balanced binary tree scheme that guarantees logarithmic worst case performance and an internal path length substantially smaller than the path length of AVL trees?

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] A. Brüggemann-Klein and D. Wood. *Drawing Trees Nicely with TEX.* Technical Report , Department of Computer Science, University of Waterloo, 1987.

[3] C.C. Foster. Information storage and retrieval using AVL trees. In *Proceedings of the ACM National Conference*, pages 192–205, 1965.

[4] G. Gonnet. Balancing binary trees by internal path reduction. *Communications of the ACM*, 26:1074–1081, 1983.

[5] R. Klein and D. Wood. The node visit cost of brother trees. *Information and Control*, 1987.

[6] R. Klein and D. Wood. *On the Path Length of Binary Trees.* Technical Report , Department of Computer Science, University of Waterloo, 1987.

[7] D.E. Knuth. *The Art of Computer Programming, Vol.3: Sorting and Searching.* Addison-Wesley Publishing Co., Reading, Mass., 1973.

[8] 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.

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

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

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