CIS-TR-80-13

# HOW UNOPTIMAL ARE AVL SEARCH TREES?

by

Sandra Mitchell Hedetniemi*

## Abstract

The following two questions can be asked about AVL search trees:

(1) Given any integer k, does there exist an AVL tree with two leaves at level numbers differing by k?

(2) What is the minimum number of nodes such a tree can have?

It is shown that a subset of the Fibonacci search trees have the minimum number of nodes among all AVL search trees with two leaves at level numbers differing by k.

* Department of Computer and Information Science, University of Oregon, Eugene, OR 97403.

## 1. Introduction

Binary search trees are well recognized as a means of representing and handling symbol tables and file directories. File directories, in particular, are characterized by numerous insertions and deletions, requiring constant updating to the data structure. The optimal binary search tree, with file names at the leaves, is the complete binary tree (all leaves occur at the same level) or the extended binary tree (all leaves occur at most two levels). These trees are "balanced" since the searches in these trees are all the same length (plus or minus one question). Various procedures have been proposed for maintaining some resemblance of balance in the search tree despite the updates; these include height-balanced (AVL [1] or Foster's generalized version [5]), weight-balanced [2], bounded-balance [8], brother trees [9] and power trees [7].

Baer and Schwab have performed comparative studies on several of these balancing procedures. They recommend using AVL trees if the number of queries per insertion is greater than 3 but less than 8. It is this class of trees which we characterize further.

## 2. AVL search trees

Let a binary tree T be defined by a tuple $< T_l, 1, T_r >$ where $T_l$ designates the left subtree and $T_r$ designates the right subtree. Define the level $l_i$ of vertex i in T to be 1 if i is the root of T or one plus the level of its parent otherwise. The level of a leaf of T corresponds to the number of questions that must be asked in order to find the leaf during a search, i.e. the length of the path followed in searching. Measures performed on the above-mentioned balancing algorithms are typically given in terms of maximum (worst-case) path length or average (expected) path length.

AVL trees, named for Adel'son-Vel'skiy and Landis, are designed to restrict the heights of the subtrees of each node. The height of a node is zero for a nonexistent node in the tree or one greater than the maximum of the heights of all the children of the node. The height of a subtree is the height of the root of that subtree. A binary search tree is an AVL tree if the difference between the heights of the left and right subtrees of each node is either -1, 0, or 1. Figure 1 illustrates two AVL trees; the first is also an extended binary tree.

Figure 1.

Figure 1(b) leads one to ask the following interrelated questions about "how far from optimal" an AVL tree can be:

(1) Given any integer k, does there exist an AVL tree with two leaves at level numbers differing by k?

(2) What is the minimum number of nodes such a tree can have?

Let us reconsider the definition of an AVL tree. The definition is expressed in terms of the difference of the level of a root r and the level of the deepest leaf in r's left subtree and the level of the deepest leaf in r's right subtree. This, of course, says nothing about the level of any other leaves in these two subtrees. However, since the definition asserts that this restriction is true at each node, these other leaves will have their level restricted because they are also in the subtree of another node which is a descendant of r. It is these leaves which help to produce the differences in levels as desired in Question (1).

In order to describe those AVL trees which satisfy questions (1) and (2), we first define the class of binay search trees known as Fibonacci trees. These trees were first used as a searching technique by Ferguson [4]. The Fibonacci trees are so named because their construction is similar to the construction of the well-known Fibonacci sequence 0,1,1,2, 3,5,8, ... where each term in the sequence is the sum of the previous two. Formally, the i-th Fibonacci number $\phi_i = \phi_{i-1} + \phi_{i-2}$, for $i \geq 3$. If we let $F_0$ denote the empty binary tree, and $F_1$ denote the binary tree with one node, then the (i+1)-st Fibonacci tree $F_i$ is defined to be $\langle F_{i-2}, 1, F_{i-1} \rangle$. It has previously been shown in [6] that a Fibonacci tree of height h, $F_h$, has the fewest nodes among all those AVL trees having height h. We show that a subset of the Fibonacci trees are those AVL trees which have leaves at level numbers differing by k, for any k, and which also have the fewest number of nodes.

We first present an alternative constructive definition of $F_h$ which shows a closer relationship to the Fibonacci numbers. $F_h$ can be constructed from $F_{h-1}$ by joining a new leaf to every node in $F_{h-1}$ which does not have 2 children. The number of leaves added to $F_{h-1}$ will be $\phi_{h-1} + \phi_{h-2}$, where $\phi_{h-1}$ denotes the number of leaves in $F_{h-1}$ and $\phi_{h-2}$ denotes the number of nodes with one child. This alternative definition more clearly demonstrates that a Fibonacci tree is an AVL tree.

We will use this second definition to show the minimality of the Fibonacci trees in meeting the condition of the difference in the levels of the leaves. The first seven Fibonacci trees are presented in Figure 2. Some of the nodes have been labeled to help identify the construction according to the second definition. In particular, nodes d, e, and f have been added to nodes a, b and c, respectively, in $F_3$ to create $F_4$. Nodes g, h, i, j and k have been added to nodes a, d, e, c and f, respectively, in $F_4$ to create $F_5$.

Figure 2

We see that the differences between the minimum and maximum level of leaves is 1 in both $F_3$ and $F_4$, but is 2 in $F_5$. We consider the data in Table 3 to predict when this change will occur in larger Fibonacci trees. We denote a node with only one child as a single parent, sp. The minimum level of a leaf in $F_h$ is

$$1 + \min \{ \min \{l_v\} , \min \{l_{sp}\} \},$$

where v is taken over all leaves in $F_{h-1}$ and sp is taken over all single parents in $F_{h-1}$. But

$$\min \{l_{sp}\} = \min \{l_w\},$$

where w is taken over all leaves in $F_{h-2}$ when $h \geq 5$. This follows since a leaf in $F_{h-2}$ will take two iterations to create both its children (where its children will have the same level in the tree).

Table 3

Since the minimum level is incremented only every other iteration and the maximum level is incremented every iteration, the difference between these two levels in $F_h$ is $\left\lfloor \frac{i-1}{2} \right\rfloor$, where i is the number of iterations required to create $F_h$ from $F_0$. But that means i is just h.

Thus, it follows from the construction that $F_{2k-1}$ is an AVL tree with two leaves at level numbers differing by k. Furthermore, $F_3$ and $F_5$ are clearly the AVL trees with the minimum number of nodes for which this is true when k = 1 or 2. By induction, using the second construction of the Fibonacci trees, this can be shown true for all k. The number of nodes in $F_{2k-1}$ is easily shown to be:

$$\sum_{i=1}^{2k+1} \Phi_i \quad .$$

Acknowledgement: The author would like to thank Reed Lacy for the suggestion of question 1.

## 3. References

[1] Adel'son-Vel'skiy, G. and Landis, Y., An algorithm for the organization of information, Doklady Akad. Nauk, SSSR(Moscow), 16(1962), 236-266. Translation, OTS, JPRS, 17, 137, Dept. of Commerce, Washington, D. C.

[2] Baer, J.-L., Weight-balanced trees, Proc. AFIFS 1975 NCC, Vol. 44, AFIFS Press, Montvale, N. J., 467-472.

[3] Baer, J.-L. and Schwab, B., A comparison of tree-balancing algorithms, Comm. ACM, 20(1977), 322-331.

[4] Ferguson, D., Fibonaccian searching, Comm. ACM, 3(1960), 648.

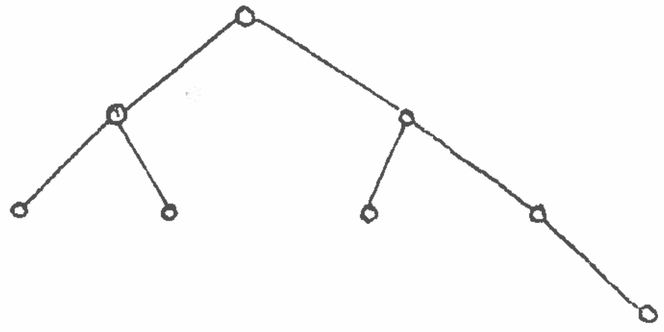
[5] Foster, C., A generalization of AVL trees, Comm. ACM, 16(1973), 512-517.

[6] Knuth, D., The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, Reading, Mass., 1973.

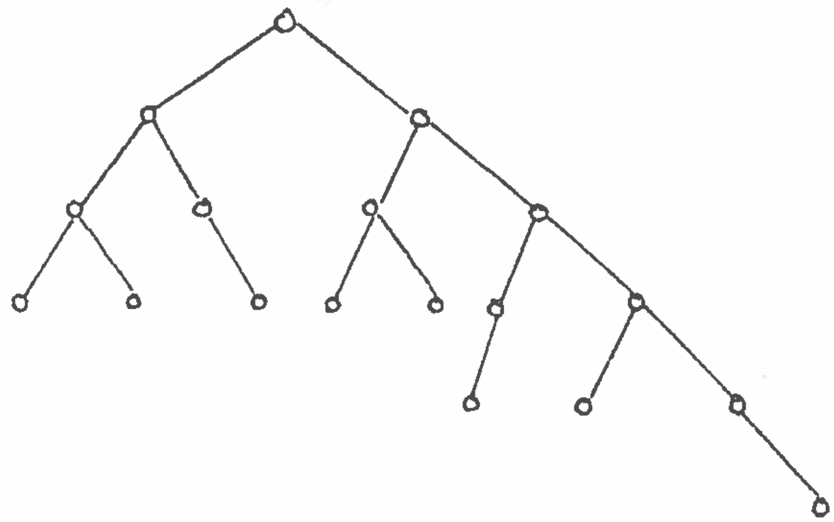
[7] Luccio, F. and Pagli, L., Power Trees, Comm. ACM, 21(1978), 941-946.

[8] Nievergelt, J. and Reingold, E., Binary search trees of bounded-balance, SIAM J. Comput., 2(1973), 33-43.

[9] Ottman, Th., Six, H. and Wood, D., Right brother trees, Comm. ACM, 21(1978), 769-776.



(a)



(b)

Figure 1

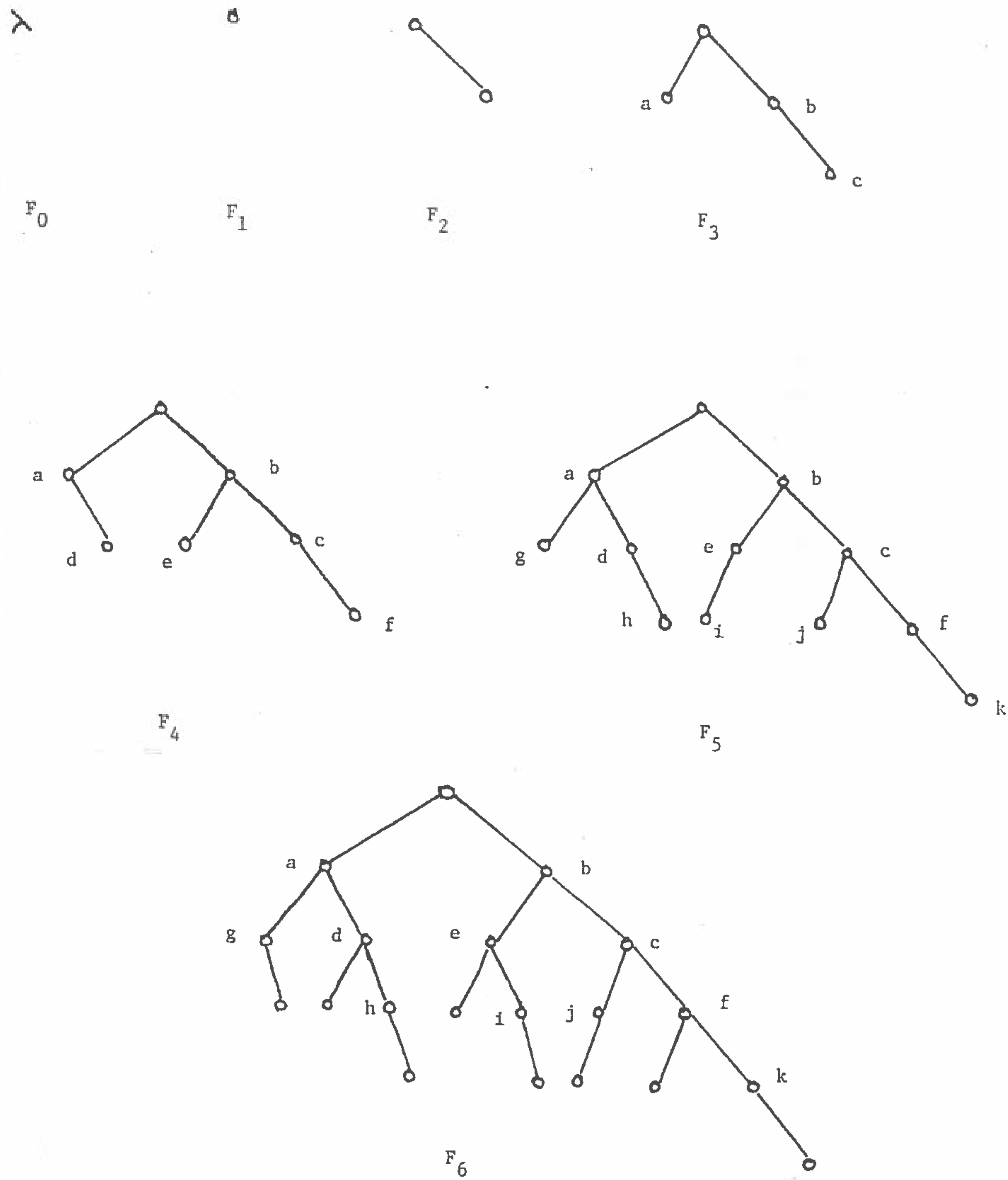



Figure 2

| | Min level of leaf in $T_l$ | Min level of single parent | Max level of leaf in $T_r$ | Difference min and max leaf's |
|---|---|---|---|---|
| $F_2$ | | 1 | 2 | |
| $F_3$ | 2 | 2 | 3 | 1 |
| $F_4$ | 3 | 2 | 4 | 1 |
| $F_5$ | 3 | 3 | 5 | 2 |
| $F_6$ | 4 | 3 | 5 | 2 |
| $F_7$ | 4 | 4 | 7 | 3 |

Table 3