An Analysis of Two Open Problems of AVL Tree Insertion Algorithm

NINGPING SUN†,RYOZO NAKAMURA††, WENLING SUN†‡,AKIO TADA‡‡
†Department of Information and Computer Science,
Kumamoto National College of Technology,Japan
††Kumamoto University,Japan
†‡Central University for Nationalities,China
‡‡Sojo University,Japan

2659-2 Suya,Nishigoshi, Kikuchi,Kumamoto 861-1102, JAPAN
Phone:+81-096-242-6104 Fax: +81-096-242-6106
sningping@cs.knct.ac.jp http://www.cs.knct.ac.jp/~sun/sun.html

Abstract: Two open problems of the AVL tree insertion algorithm are: 1. If all $n!$ permutations of n keys occur with the equal probability, what is the expected height of the constructed AVL tree? 2. What is the rebalancing probability due to an insertion ? In order to analyze these problems, an approach is proposed to count the number of the permutations constructed AVL tree with a given number of nodes and height.Consequently, the formulae are derived to compute the expected height of AVL trees and the rebalancing probability.The numerical test results are presented finally.

Key-Words: Analysis of algorithm,AVL tree and permutation,AVL tree insertion algorithm,Expected height of tree,Rebalancing probability

1 Introduction

The AVL tree technique named after its inventors, G.M.Adel'son-Vel'skii and E.M.Landis,is one of the best known algorithms to balance a binary search tree. AVL trees obey the following rule: *The heights of the child trees of any given node differ by at most one.* In this paper, the height of a tree is defined to be the length of the longest path from its root to one of its leaf nodes.

The mathematical analysis of the following two problems about AVL tree insertion algorithm have not been done successfully.

1. *If all $n!$ permutations of n keys occur with the equal probability, what is the expected height of the constructed AVL balanced tree?*
2. *What is the probability that an insertion requires rebalancing?*

As an important step to the analysis,we first propose an approach to count the number of the permutations constructed the AVL tree with a given number of nodes and height.

2 AVL Tree and its Permutations

The height of an AVL tree with n internal nodes,h_n,has been proved is to be

$$\lceil \mathrm{lg(n+1)} \rceil - 1 \leq \mathrm{h_n} \leq 1.4404 \lg(\mathrm{n}+1) - 1.328. \quad (1)$$

The upper bound of Eq.(1) is the height of Fibonacci tree and the lower bound is the height of perfectly balanced tree.

Given a number of nodes,n, Fibonacci tree is the highest AVL tree with n nodes.The height of Fibonacci tree with n nodes can be expressed by

$$H_n = \begin{cases} i-3, & if \quad n = F_i - 1 \\ & \quad (n \geq 1, i \geq 3); \\ i-4, & if \quad F_{i-1} - 1 < n < F_i - 1 \\ & \quad (n \geq 3, i \geq 5), \end{cases} \quad (2)$$

where F_i means the Fibonacci number of order i,

$$F_i = F_{i-1} + F_{i-2}$$

for $i \geq 2$ with $F_0 = 0, F_1 = 1$.

The closed form of F_i is

$$F_i \doteq \phi^i/\sqrt{5}, \quad \phi = (1+\sqrt{5})/2.$$

On the other hand,given a height of tree,h, Fibonacci trees have the minimum number of nodes. Let f_h be the number of nodes of the Fibonacci tree with height of h,

$$f_h = f_{h-1} + f_{h-2} + 1,$$

for $h \geq 2$ with $f_0 = 1, f_2 = 2$.

Asymptotically,

$$f_h = F_{h+3} - 1 = \lfloor \phi^{h+3}/\sqrt{5} - 1 \rfloor, for \ \ h \geq 0. \quad (3)$$

On the contrary, the perfectly balanced trees have the maximum number of nodes,$2^{h+1}-1$.

Fig.1 illustrates the structure of the AVL tree with n nodes and height h.

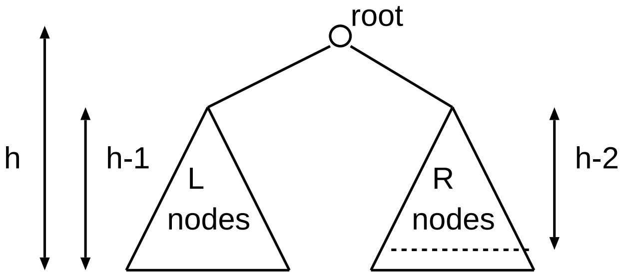


Fig.1 The Structure of the AVL tree with n nodes($n = L + R + 1$) and height h

For the root,if its left subtree is an AVL tree with L nodes and height $h-1$,its right subtree must be constructed as an AVL tree with R nodes ($R = n - L - 1$) and height $h-1$ or $h-2$,and vice versa.

We consider the range of L,the number of nodes of the left subtree. From above arguments, the lower bound of L will be f_{h-1}. The left subtree of height $h-1$ must have no more than $2^h - 1$ nodes, and the height of the right subtree has no less than $h-2$. Therefore,the upper bound of L should be $\min\{2^h - 1, n-1-f_{h-2}\}$. $r(L_{h-1})$ stands for

$$f_{h-1} \leq L \leq \min\{2^h - 1, n-1-f_{h-2}\}$$

.

If all $n!$ permutations of n keys occur with the equal probability, which of these permutations could build an AVL tree without rebalancing operations? For instance,among 6 possible permutations of 3 nodes $(123, 132, 213, 231, 312, 321)$ only 213 and 231 are the permutations that will construct AVL tree without rebalance. We call them **AVL tree permutations** for convenience.

Assuming that in an AVL tree with n nodes and height h as shown in Fig.1, the left subtree of height h_L, ($h_L = h-1$), has L nodes $\{a_1, \cdots, a_L\}$, a permutation of these nodes is $a_1 \cdots a_L$; the right subtree of height h_R,($h_R = h-1$ or $h_R = h-2$), has R nodes $\{b_1, \cdots, b_R\}$, a permutation of these nodes is $b_1 \cdots b_R$, illustrated in Fig.2.

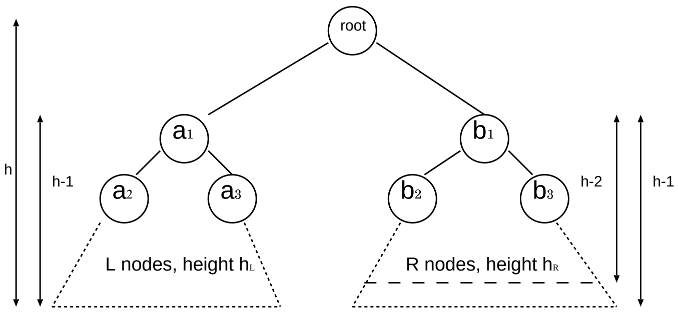
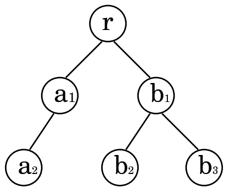


Fig.2 The Structure of the AVL tree

The following definition are made in order to find AVL tree permutations consisting of $a_1 \cdots a_L$ and $b_1 \cdots b_R$. Within the permutation $a_1 \cdots a_L$, we call the left position of a_i the $(i-1)$th position and the right position of a_i the ith position $(i = 1, \ldots, L)$; the most left position of $a_1 \cdots a_L$ the 0th position and the most right position the Lth position.Let k_i be the number of objects of $b_1 \cdots b_R$ that could be arranged from the 0th to the ith position of $a_1 \cdots a_L$,namely, when the left subtree is constructed with i nodes under the AVL rule,k_i is the number of nodes of the corresponding right subtree.Meanwhile, $(k_i - k_{i-1})$ means the number of objects of the $b_1 \cdots b_R$ that could be arranged in the ith position of $a_1 \cdots a_L$.

In the following example,we can see that the AVL tree permutations shown in the right of the following figure, are actually the arrangements consisting of $a_1 \cdots a_L$ and $b_1 \cdots b_R$ with AVL rule, followed the root node.



r a_1 b_1 a_2 b_2 b_3
r a_1 b_1 b_2 a_2 b_3
r a_1 b_1 b_2 b_3 a_2
r b_1 a_1 b_2 a_2 b_3
r b_1 a_1 b_2 b_3 a_2
r b_1 a_1 a_2 b_2 b_3

In an AVL tree permutation how to arrange b's into each position of $a_1 \cdots a_L$, in other words,what is the range of k_i? First, we consider min_i,the lower bound of k_i. Inserting the $(i+1)$th node into the left subtree might make the left subtree taller,the left subtree probably becomes a Fibonacci tree with $(i+1)$ nodes and height H_{i+1},the height of the left subtree is $\min\{H_{i+1}, h_L\}$. The right subtree must be no lower than $\min\{H_{i+1}, h_L\} - 1$. Therefore, the minimum number of k_i is

$$min_i = f_{\min\{H(i+1),h_L\}-1}. \quad (4)$$

where $1 \leq i \leq L$.

Secondly, we consider max_i, the upper bound of k_i. If the left subtree is an AVL tree with i nodes and height H_i, the right subtree must have no more than $2^{\min\{h_L,H_i\}+2} - 1$ nodes. Therefore,the maximum number of k_i is

$$max_i = \min\left\{R, 2^{\min\{h_L,H_i\}+2} - 1\right\}. \quad (5)$$

where $1 \leq i \leq L$.

G_{L,h_L,R,h_R} denotes the number of arrangements consisting of $a_1 \cdots a_L$ and $b_1 \cdots b_R$.These arrangements obey the balance criterion of AVL tree.

$$\begin{aligned} G_{L,h_L,R,h_R} &= 2 \sum_{k_1=min_1}^{max_1} \binom{k_1-k_0}{k_1-k_0} \\ &\cdot \sum_{k_2=min_2}^{max_2} \binom{k_2-k_1}{k_2-k_1} \cdots \sum_{k_L=R}^{R} \binom{k_L-k_{L-1}}{k_L-k_{L-1}} \quad (6) \end{aligned}$$

where $1 \leq i \leq L$.

G_{L,h_L,R,h_R} is a key to our analysis. We could get the number of AVL tree permutations with a given number of nodes and height as follows.

$$\begin{aligned} PE_{n,h} &= \sum_{r(L_{h-1})} PE_{L,h-1}\{G_{L,h-1,R,h-1} PE_{R,h-1} \\ &+ 2G_{L,h-1,R,h-2} PE_{R,h-2}\} \quad (7) \end{aligned}$$

for $n > 3$ and $h \geq 2$ with $PE_{1,0} = 1, PE_{2,1} = 2, PE_{3,1} = 2$.

Fig.3 gives some results from Eq.(7).From these results we can see that for a given height of tree,h, when n is increasing from f_h, the value of $PE_{n,h}$ is increasing steeply.

Moreover,the number of the AVL tree permutations with n nodes can be expressed by

$$PE_n = \sum_h PE_{n,h} \quad (8)$$

where $\lceil \lg(n+1) \rceil - 1 \leq h \leq 1.44 \lg(n+1) - 1.328$.

We are interested in the percentage of that AVL tree permutations with n nodes occur among all $n!$ permutations of n nodes.From Table 1 and Fig.4,this percentage is extremely low: when $n = 10$ it is 0.8%,and when $n \geq 20$ it nears to 0%.

Table 1: Some results of Eq.(8) and $n!$

n	PE_n	$n!$
1	1	1
2	2	2
3	2	6
4	8	24
5	24	120
6	48	720
7	240	5040
8	1152	40320
9	5760	362880
10	29952	3628800
11	145152	39916800
12	857088	479001600
13	6165504	6227020800
14	46614528	87178291200
15	383698944	1307674368000
16	3361554432	20922789888000
17	30556569600	355687428096000
18	285372481536	6402373705728000
19	2709051899904	121645100408832000
20	26614178906112	2432902008176640000

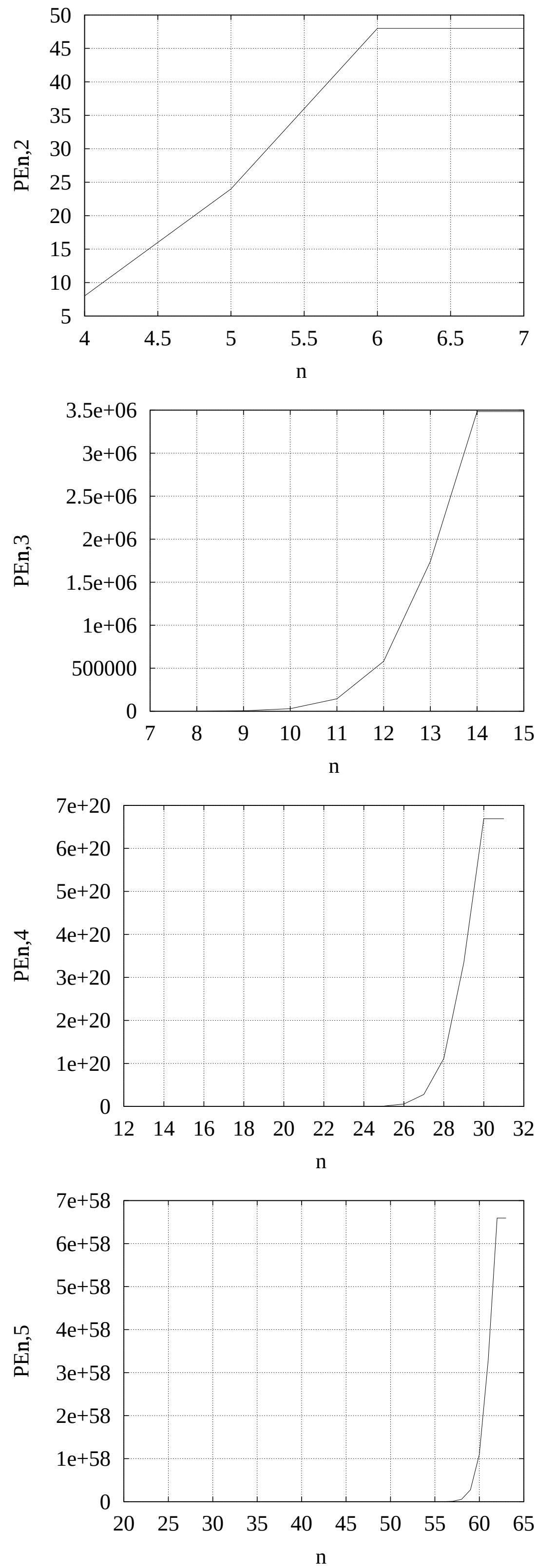


Fig.3 The number of the AVL tree permutations with a given number of nodes and height

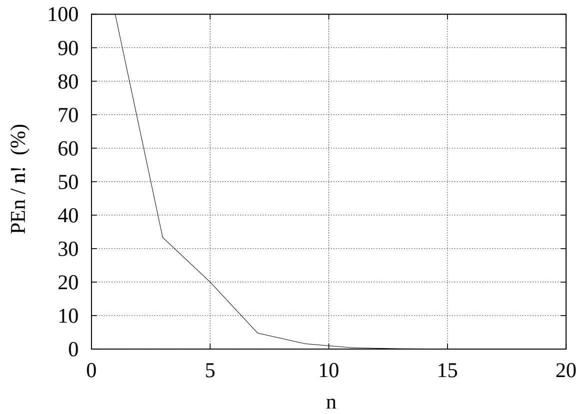


Fig.4 The percentage of AVL tree permutations to $n!$

3 The Analysis of Two Open Problems

Now we analyze the two open problems of AVL tree insertion algorithm based on the above analysis. First, the expected height of AVL trees can be obtained by

$$HAVL(n) = \sum_h hPE_{n,h}/PE_n, \qquad (9)$$

Fig.5 gives results from Eq.(9).The expected hight of the AVL tree with n nodes approximates to $1.21 \lg n - 0.7$.

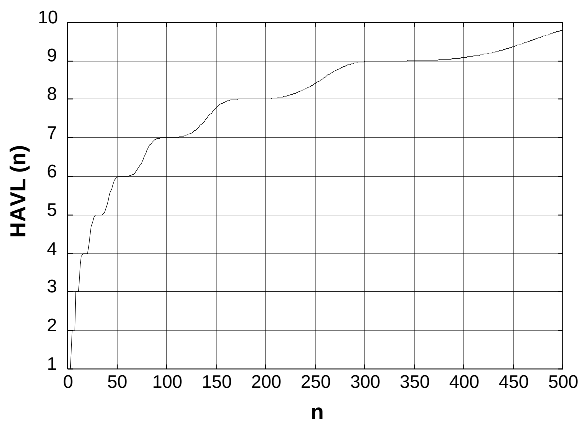


Fig.5 The expected height of AVL trees with n nodes

To the second open problem,we have revealed that there are only two different patterns in AVL trees, *pattern-1* and *pattern-2*,associated with the imbalance([1]). They are shown in Fig.6 (a) where the nodes added by insertions are indicated by crosses. Based on these two proposed patterns, single and double rotations are indicated by two essentially different types of subtrees in Fig.6(b),where S stands for a single rotation and D for a double rotation.

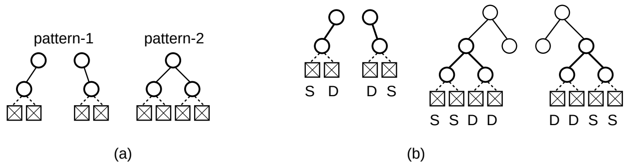


Fig. 6 Two essential patterns in an AVL tree

Obviously,an insertion at the external nodes of the leaf of pattern-1 should cause imbalance as shown in Fig.7 (a). Differently, inserting a new key into the external nodes of pattern-2 could cause imbalance or not. Only the pattern-2's in the subtrees which are taller than their adjacent subtrees are associated with imbalance,such as those drawn within the dotted square in the Fig.7 (b).

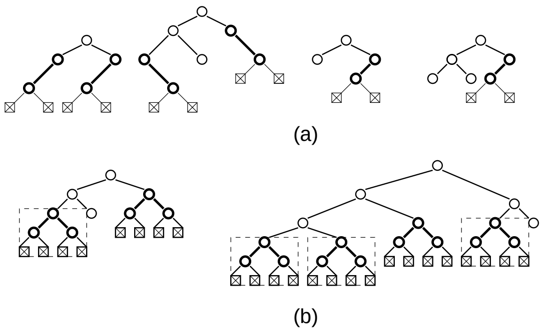


Fig. 7 The instances of imbalance associated with pattern-1 and pattern-2

When all $n!$ permutations of n nodes occur with equal probability,$p_{1_{n,h}}$ denotes the average number of pattern-1 and $p_{2_{n,h}}$ denotes the average number of the pattern-2 that will cause imbalance among the AVL tree permutations with n nodes and height h respectively.

$$\begin{aligned} p_{1_{n,h}} &= \frac{1}{PE_{n,h}} \sum_{r(L_{h-1})} PE_{L,h-1} \\ &\cdot[G_{L,h-1,R,h-1} PE_{R,h-1}\{p_{1_{L,h-1}} + p_{1_{R,h-1}}\} \\ &+2G_{L,h-1,R,h-2} PE_{R,h-2}\{p_{1_{L,h-1}} + p_{1_{R,h-2}}\}] \end{aligned} \tag{10}$$

for $n \geq 4$ with $p_{1_{2,1}} = 1, p_{1_{3,1}} = 0$.

And,

$$\begin{aligned} p_{2_{n,h}} &= \frac{1}{PE_{n,h}} \sum_{r(L_{h-1})} PE_{L,h-1} \\ &\cdot[G_{L,h-1,R,h-1} PE_{R,h-1}\{p_{2_{L,h-1}} + p_{2_{R,h-1}}\} \\ &+2G_{L,h-1,R,h-2} PE_{R,h-2}\{p_{3_{L,h-1}} + p_{2_{R,h-2}}\}] \end{aligned} \tag{11}$$

for $n \geq 5$ and $n \geq 2$ with $p_{2_{5,2}} = 1/3$.

Here,$p_{3_{n,h}}$ is the indirectly recursive function of $p_{2_{n,h}}$.

$$\begin{aligned} p_{3_{n,h}} &= \frac{1}{PE_{n,h}} \sum_{r(L_{h-1})} PE_{L,h-1} \\ &\cdot[G_{L,h-1,R,h-1} PE_{R,h-1}\{p_{3_{L,h-1}} + p_{3_{R,h-1}}\} \\ &+2G_{L,h-1,R,h-2} PE_{R,h-2}\{p_{3_{L,h-1}} + p_{2_{R,h-2}}\}] \end{aligned} \tag{12}$$

for $n \geq 3$ and $n \geq 1$ with $p_{3_{3,1}} = 1$.

As shown in Fig.6(b), the single rotation and the double rotation will happen equally likely. For a rotation, pattern-1 has one,and pattern-2 has two inserting positions. We use $PR(n,h)$ to express the probability that a single or a double rotation should be used to restore the desired balance when a new node is inserted into the AVL tree with n nodes and height h.

$$PR(n,h) = \frac{1}{n+1} p_{1_{n,h}} + \frac{2}{n+1} p_{2_{n,h}}. \tag{13}$$

Finally, when the $(n+1)$th nodes is inserted into the AVL tree with n nodes, the rebalancing probability is

$$P(n) = \frac{1}{PE_n} \sum_h PE_{n,h} PR(n,h), \tag{14}$$

where $\lceil \lg(n+1) \rceil - 1 \leq h \leq 1.44 \lg(n+1) - 1.328$.

From the results of Fig.8 and 9, we can see that when $n \geq 20$,the rebalancing probability occurs during insertion operations approximates to 0.28.Because every AVL tree has its mirrored one, one insertion needs 0.56 rotations on average.

4 Conclusions

In order to analyze the two problems of AVL tree insertion algorithm,we have proposed a reccurence method. The recursive description about AVL tree and its permutations is the key to our analysis. However,the larger the number of nodes, the more complicated the computing of proposed formulae. In our future work we will derive an asymptotical expression of these formulae.

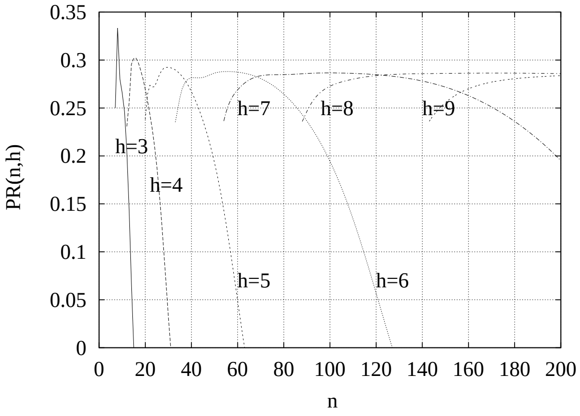


Fig.8 Some results of $PR(n, h)$

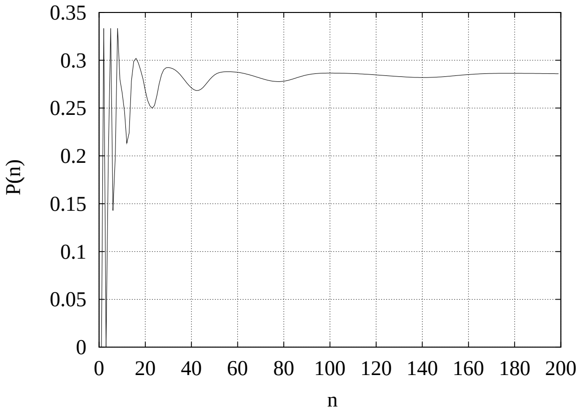


Fig.9 Some results of $P(n)$

References

[1] R. Nakamura, N. Sun, T. Nakashima: An Approximate Analysis of the AVL Balanced Tree Insertion Algorithm,*Trans. IPS. Japan*, Vol.39,No.4,pp.1006-1013,1998.

[2] D. E. Knuth:*The Art of Computer Programming*,Vol.3, Sorting and Searching, Addison-Wesley,pp.458-470,1998.

[3] N. Wirth: *Algorithms + Data Structures = Programs*,Prentice- Hall,pp.216-226,1976.

[4] G.M. Adel'son-Vel'skii, E. M. Landis: An Algorithm for the Organization of Information, *English trans. in Soviet Math*, Vol.3,pp.1259-1263,1962.

[5] D.H.Greene,D.E.Knuth: *Mathematics for the Analysis of Algorithm,* Birkhäuser,pp.30,1990.

[6] R.Sedgewick,P.Flajolet: *An Introduction to the Analysis of Algorithms,* Addison-Wesley,1996