Acta Informatica 9, 159 – 170 (1978)

Acta Informatica

On Random 2–3 Trees *

Andrew Chi-Chih Yao

Computer Science Department, Stanford University, Stanford, CA 94305, USA

Summary. It is shown that $\bar{n}(N)$, the average number of nodes in an N-key random 2–3 tree, satisfies the inequality $0.70\,N < \bar{n}(N) < 0.79\,N$ for large N. A similar analysis is done for general B-trees. It is shown that storage utilization is essentially $\ln 2 \approx 69\,\%$ for B-tree of high orders.

1. Introduction

Balanced tree structures are often used in the organization of information. One attractive scheme, called "2–3 trees", was introduced by J. Hopcroft [1] [5, pp. 468–471]. Some interesting questions concerning 2–3 trees have been raised [3]. In this paper we present a partial solution to a problem posed in [3].

A 2–3 tree is a tree in which every internal node contains either 1 or 2 keys, and all the leaves are at the same level (see Fig 1). In drawing 2–3 trees we shall adopt the notation used in [3]. Thus keys are represented by dots inside a node as we shall only be interested in the structure of the trees.

To put a new key into a node that contains only one key, we simply insert it as a second key. If the node already contains 2 keys, we *split* the node into two nodes containing respectively the minimum and the maximum of the three keys, and insert the middle key into the parent node by repeating the process. When there is no node above, a new root node will be created to hold the middle key.

Consider a 2–3 tree T with $j-1$ keys in it. These $j-1$ keys divide all possible key values into j intervals. The insertion of a new key K_j into T is said to be a *random insertion* if K_j has equal probabilities for being in any one of the j intervals defined above.

Now consider the building of 2–3 trees by successive random insertions. The average cost involved is dependent on the specific implementation of the

* This work was done while the author was at University of Illinois, partially supported by NSF Grant GJ 41538. The preparation of this paper has also been partially supported by NSF Grants MCS 72-06336 A04 and MCS 72-03752 A03

0001-5903/78/0009/0159/$02.40

insertion algorithm. There are, however, certain quantities that are useful in general for the analysis. One quantity of interest is $\bar{n}(N)$, the average number of internal nodes in a 2–3 tree after N keys have been randomly inserted into the empty tree. In this paper we shall derive bounds on $\bar{n}(N)$ and on the corresponding quantity for *B-trees* (see Section 3). A systematic procedure for deriving improved bounds is discussed, but the computation involved appears to be prohibitive. Main results are contained in Theorems 2.7, 2.12 and 3.1.

We shall use the term *N-key random 2–3 tree* for a 2–3 tree obtained by N random insertions. It is easy to show (by induction on N) that an equivalent definition is a 2–3 tree obtained by successively inserting N keys $K_1, K_2, \ldots, K_N$ into the empty tree, assuming each of the $N!$ linear orderings of the keys are equally likely. Thus, for example, a random 2–3 tree may be built by drawing the K_j's independently from a common continuous distribution.

2. Number of Nodes in 2–3 Trees

Let T be any 2–3 tree. We shall use $n(T)$ to denote the number of internal nodes of T. Let $f_N(T)$ be the probability that T will result after N random insertions are made to the empty tree. Obviously $f_N(T)$ is zero unless T contains exactly N keys. In terms of $f_N(T)$ and $n(T)$, the average number of nodes $\bar{n}(N)$ defined in Section 1 can be expressed as follows:

$$\bar{n}(N)=\sum_T n(T)\, f_N(T). \tag{1}$$

To derive bounds on $\bar{n}(N)$, we observe that most of the internal nodes of any 2–3 tree appear on the lowest few levels. Therefore, a good estimate of $\bar{n}(N)$ can be obtained by analyzing the number of internal nodes in those levels of a random 2–3 tree. We shall carry out the analysis for the lowest level first in the next subsection, and then take the second lowest level into account in Section 2.2.

2.1. First Order Analysis

As shown in Figure 2, there are two types of 2–3 trees of height 1; the type 1 tree contains 1 key and the type 2 tree contains 2 keys. An arbitrary 2–3 tree T is said to be of *class* $(1; x_1, x_2)$ if exactly x_1 of T's height 1 subtrees are of type 1 and x_2 are of type 2. The tree shown in Figure 1 is of class $(1; 3, 2)$.

Let T be an N-key 2–3 tree of class $(1; x_1, x_2)$. The following lemmas are easy to obtain:

Lemma 2.1. $2x_1+3x_2=N+1$.

Proof. Both $N+1$ and $2x_1+3x_2$ are equal to the number of leaves of T. □

Lemma 2.2. $\frac{3}{2}(x_1+x_2)-\frac{1}{2}\leqq n(T)\leqq 2(x_1+x_2)-1$.

Proof. There are x_1+x_2-1 keys contained in the internal nodes above the lowest level. Thus the number of nodes above the lowest level, $n(T)-(x_1+x_2)$, satisfies $\frac{1}{2}(x_1+x_2-1)\leqq n(T)-(x_1+x_2)\leqq x_1+x_2-1$. Lemma 2.2 follows. □

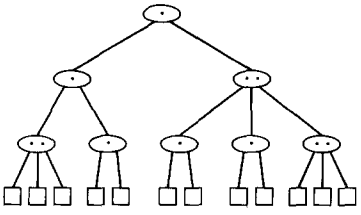

Fig. 1. A 2-3 tree with 11 keys

Fig. 2. The two types of 2-3 trees of height 1

Definition 2.3. Let $\mathscr{F}(x_1, x_2)$ be the set of 2-3 trees of class $(1; x_1, x_2)$. Define $P_N(x_1, x_2) = \sum_{T \in \mathscr{F}(x_1, x_2)} f_N(T)$, the probability that a tree of class $(1; x_1, x_2)$ results after N random insertions.

Definition 2.4. $A_i(N) = \sum_{x_1, x_2} x_i \cdot P_N(x_1, x_2)$, $i = 1, 2$. Thus, $A_i(N)$ is the average value of x_i for random N-key 2-3 trees.

Lemma 2.5. $\frac{3}{2}(A_1(N) + A_2(N)) - \frac{1}{2} \leqq \bar{n}(N) \leqq 2(A_1(N) + A_2(N)) - 1$.

Proof. This follows from Lemma 2.2 and the definitions of $\bar{n}(N)$, $A_1(N)$, $A_2(N)$. □

Lemma 2.6. $A_1(N) = \frac{2}{7}(N+1)$, $A_2(N) = \frac{1}{7}(N+1)$ for $N \geqq 6$.

Proof. Let T be an $(N-1)$-key 2-3 tree of class $(1; x_1, x_2)$. By making a random insertion into T, we will obtain a 2-3 tree either of class $(1; x_1 - 1, x_2 + 1)$ or of class $(1; x_1 + 2, x_2 - 1)$. The former situation happens, with probability $2x_1/N$, when the new key is inserted into a subtree of type 1. Thus we have

$$\begin{aligned} A_1(N) &= \sum_{x_1, x_2} P_{N-1}(x_1, x_2) \left(\frac{2x_1}{N}(x_1 - 1) + \left(1 - \frac{2x_1}{N}\right)(x_1 + 2) \right) \\ &= \sum_{x_1, x_2} P_{N-1}(x_1, x_2) \left(x_1 + \frac{-6x_1}{N} + 2 \right) \\ &= \left(1 - \frac{6}{N}\right) A_1(N-1) + 2. \end{aligned} \tag{2}$$

With initial condition $A_1(1) = 1$, it is easy to show from (2) that

$$A_1(N) = \tfrac{2}{7}(N+1) \quad \text{for } N \geqq 6. \tag{3}$$

Lemma 2.1 implies $2A_1(N) + 3A_2(N) = N + 1$. This and (3) give

$$A_2(N) = \tfrac{1}{7}(N+1) \quad \text{for } N \geqq 6. \quad \square$$

Lemmas 2.5 and 2.6 lead immediately to the following theorem:

Theorem 2.7. $\frac{9}{14}N + \frac{1}{7} \leqq \bar{n}(N) \leqq \frac{6}{7}N - \frac{1}{7}$ for $N \geqq 6$.

Corollary. $0.64\,N \leqq \bar{n}(N) \leqq 0.86\,N$ for $N \geqq 6$.

The above bounds should be compared with the obvious bounds $\frac{N}{2} \leqq \bar{n}(N) \leqq N$, which can be regarded as the zero-th order approximation of $\bar{n}(N)$.

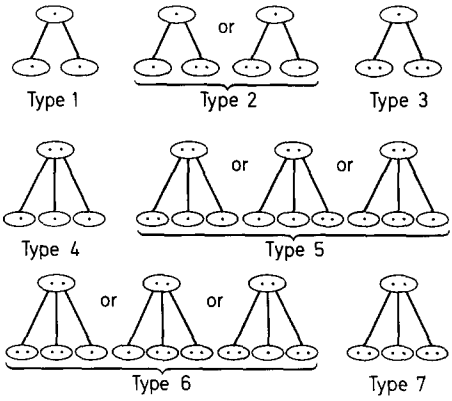

Fig. 3. There are 12 distinct 2–3 trees of height 2, classified into 7 types (leaves not shown)

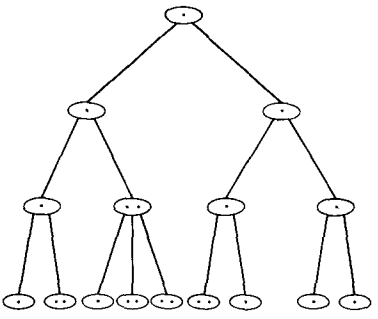

Fig. 4. A 2–3 tree of class (2; 1, 2, 0, 0, 0, 1, 0)

2.2 Second Order Analysis

Better bounds for $\bar{n}(N)$ can be derived by considering the internal nodes on the lowest 2 levels of 2–3 trees. There are 12 possible trees of height 2, which are grouped into 7 types as shown in Figure 3. For any 2–3 tree T with no fewer than 3 keys, we can classify T by its height 2 subtrees. We shall say that T is of *class* $(2; x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ if there are x_i height 2 subtrees of type i for each i (Fig. 4). Let T be an N-key 2–3 tree of class $(2; x_1, x_2, \ldots, x_7)$. The following two lemmas are easy to prove.

Lemma 2.7. $4x_1+5x_2+6x_3+6x_4+7x_5+8x_6+9x_7=N+1$.

Proof. Similar to the proof of Lemma 2.1. □

Lemma 2.8. $\frac{7}{2}\sum_{i=1}^{3} x_i+\frac{9}{2}\sum_{i=4}^{7} x_i-\frac{1}{2}\leqq n(T)\leqq 4\sum_{i=1}^{3} x_i+5\sum_{i=4}^{7} x_i-1$.

Proof. Similar to the proof of Lemma 2.2. □

In analogy with the notation $P_N(x_1, x_2)$ defined in Section 2.1, we use $P_N(2; x_1, x_2, \ldots, x_7)$ to denote the probability for an N-key random 2–3 tree to be of class $(2; x_1, x_2, \ldots, x_7)$. For each i $(1\leqq i\leqq 7)$, define

$$A_i(N)=\sum_{x_1,\ldots,x_7} x_i\, P_N(2; x_1, \ldots, x_7).$$

Lemma 2.9. $\frac{7}{2}\sum_{i=1}^{3} A_i(N)+\frac{9}{2}\sum_{i=4}^{7} A_i(N)-\frac{1}{2}\leqq \bar{n}(N)\leqq 4\sum_{i=1}^{3} A_i(N)+5\sum_{i=4}^{7} A_i(N)-1$.

Proof. Use Lemma 2.8 and definitions of $A_i(N)$, $\bar{n}(N)$. □

We shall study the values of the $A_i(N)$'s. Once these numbers are known, Lemma 2.9 determines $\bar{n}(N)$ to within 13%.

Table 1. Transition under a random insertion: a tree of class $(2; x_1, \ldots, x_7)$ becomes a tree of class $(2; x'_1, \ldots, x'_7)$. Each row gives the values of $x'_1, \ldots, x'_7$ for a possible resulting class with its probability of occurrence in the last column

x'_1	x'_2	x'_3	x'_4	x'_5	x'_6	x'_7	Probability
x_1-1	x_2+1	x_3	x_4	x_5	x_6	x_7	$4x_1/N$
x_1	x_2-1	x_3+1	x_4	x_5	x_6	x_7	$2x_2/N$
x_1	x_2-1	x_3	x_4+1	x_5	x_6	x_7	$3x_2/N$
x_1	x_2	x_3-1	x_4	x_5+1	x_6	x_7	$6x_3/N$
x_1	x_2	x_3	x_4-1	x_5+1	x_6	x_7	$6x_4/N$
x_1+2	x_2	x_3	x_4	x_5-1	x_6	x_7	$3x_5/N$
x_1	x_2	x_3	x_4	x_5-1	x_6+1	x_7	$4x_5/N$
x_1	x_2	x_3	x_4	x_5	x_6-1	x_7+1	$2x_6/N$
x_1+1	x_2+1	x_3	x_4	x_5	x_6-1	x_7	$6x_6/N$
x_1+1	x_2	x_3+1	x_4	x_5	x_6	x_7-1	$6x_7/N$
x_1	x_2+2	x_3	x_4	x_5	x_6	x_7-1	$3x_7/N$

Consider any $(N-1)$-key 2-3 tree T of class $(2; x_1, x_2, \ldots, x_7)$. By examining the insertion process, it can be seen that there are 11 classes of trees that T might become upon the random insertion of a key. These 11 possible classes together with their probabilities of occurrence are tabulated in Table 1. Recurrence relations for the $A_i(N)$'s can be obtained from Table 1 as in Section 2.1. For example, it is easy to show that

$$A_1(N)=\sum_{x_i\text{'s}} P_{N-1}(2;x_1,\ldots,x_7)\left[x_1+\frac{4x_1}{N}\cdot(-1)+\frac{3x_5}{N}\cdot(2)+\frac{6x_6}{N}\cdot(1)+\frac{6x_7}{N}\cdot(1)\right]$$

$$=A_1(N-1)+\frac{1}{N}(-4A_1(N-1)+6A_5(N-1)+6A_6(N-1)+6A_7(N-1)).$$

Similar formulas for $A_2(N), \ldots, A_7(N)$ can also be derived. These relations can be compactly written in the following form: Let $A(N)$ be the 7-component column vector $(A_i(N))$, then

$$A(N)=\left(I+\frac{1}{N}D\right)A(N-1) \tag{4}$$

where I is the 7×7 identity matrix and D is given by

$$D=\begin{pmatrix} -4 & 0 & 0 & 0 & 6 & 6 & 6 \\ 4 & -5 & 0 & 0 & 0 & 6 & 6 \\ 0 & 2 & -6 & 0 & 0 & 0 & 6 \\ 0 & 3 & 0 & -6 & 0 & 0 & 0 \\ 0 & 0 & 6 & 6 & -7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & -9 \end{pmatrix}. \tag{5}$$

To solve $A(N)$ from (4), we define a 7-component column vector $a(N)=(a_i(N))$ by

$$a_i(N)=A_i(N)/(N+1). \tag{6}$$

In terms of $a(N)$, (4) can be written as

$$a(N)=\left(I+\frac{1}{N+1}(D-I)\right)a(N-1). \tag{7}$$

To solve (7), the following lemma is useful. The proof is omitted here, since it can be found in Knuth [5, pp. 679–680, answer to ex. 10] where recurrence relations of the form (7) are studied.

Lemma 2.10. Let G be a $p\times p$ real matrix with simple eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{p-1}$ where $\lambda_0=0$ and $\operatorname{Re}\lambda_{p-1}\leqq\operatorname{Re}\lambda_{p-2}\leqq\cdots\leqq\operatorname{Re}\lambda_1<0$. If $v(1), v(2), \ldots, v(N), \ldots$ is a sequence of p-component vectors satisfying $v(j)=\left(1+\frac{1}{j+1}G\right)v(j-1)$, then there exists a vector u such that

(i) $Gu=0$

(ii) $|(v(N))_i-u_i|<C\,N^{\operatorname{Re}\lambda_1}$ for some constant C and all i, N, where $(v(N))_i$, u_i denote the i-th component of $v(N)$ and u respectively.

Corollary. $v(N)\to u$ as $N\to\infty$.

We can now determine the asymptotic value of $a(N)$ from (7). An explicit calculation shows that the characteristic polynomial of $D-I$ is $-\lambda(\lambda+7)\cdot(\lambda^5+45\lambda^4+835\lambda^3+8175\lambda^2+42796\lambda+95892)$. The roots of the polynomial, which are eigenvalues of $D-I$, are 0, $-6.55\pm 6.25i$, -7, $-9.23\pm 1.37i$, -13.44. Thus, $D-I$ satisfies the conditions on G in Lemma 2.10. Therefore, there exists a vector $u=(u_i)$ such that

$$|a_i(N)-u_i|<C_0\,N^{-6.55} \quad \text{for some constant } C_0 \tag{8}$$

where u satisfies

$$(D-I)\,u=0. \tag{9}$$

In terms of the u_i's, we can express Lemma 2.9 as follows:

Lemma 2.11.

$$\begin{aligned}(N+1)\left(\tfrac{7}{2}\sum_{i=1}^{3}u_i+\tfrac{9}{2}\sum_{i=4}^{7}u_i\right)-\tfrac{1}{2}-CN^{-5.55}\\ \leqq\bar{n}(N)\leqq(N+1)\left(4\sum_{i=1}^{3}u_i+5\sum_{i=4}^{7}u_i\right)\\ -1+CN^{-5.55}\quad\text{for some constant } C.\end{aligned}$$

Proof. From Equations (6) and (8), we obtain

$$|A_i(N)-(N+1)\,u_i|<C'\,N^{-5.55} \quad \text{for some constant } C'. \tag{10}$$

The lemma follows immediately from (10) and Lemma 2.9. □

Now, to find the values of the u_i's, we observe that Equation (9) determines u up to a constant factor. The normalization constant can be determined as follows:

Lemma 2.7 and Equation (6) lead to the equation

$$4a_1(N)+5a_2(N)+6a_3(N)+6a_4(N)+7a_5(N)+8a_6(N)+9a_7(N)=1. \tag{11}$$

Since $a_i(N)\to u_i$ as $N\to\infty$, (11) implies that

$$4u_1+5u_2+6u_3+6u_4+7u_5+8u_6+9u_7=1 \tag{12}$$

which is the equation we need in order to determine the normalization constant. Therefore, solving Equations (9) and (12), we obtain:

$$\begin{aligned}
u_1&=414/7991 &&=0.052,\\
u_2&=396/7991 &&=0.050,\\
u_3&=912/55937 &&=0.016,\\
u_4&=1188/55937 &&=0.021,\\
u_5&=1575/55937 &&=0.028,\\
u_6&=700/55937 &&=0.013,\\
u_7&=20/7991 &&=0.003.
\end{aligned}$$

Substituting the values of the u_i's into the inequality in Lemma 2.11, we obtain the main result of this section.

Theorem 2.12. $0.70N+0.20-CN^{-5.55}\leqq\bar{n}(N)\leqq 0.79N-0.21+CN^{-5.55}$ for some constant C.

The technique used in this subsection can be used to compute bounds on higher moments of the number of nodes $n(T)$. For example, we can set up a system of recurrence relations of the form of Equation (4) for the quantities $A_{ij}(N)$ where

$$A_{ij}(N)=\sum_{x_1,\ldots,x_7} P_N(2;x_1,\ldots,x_7)\,x_i\,x_j \qquad i,j=1,2,\ldots,7.$$

Determination of the $A_{ij}(N)$'s will then lead to (by Lemma 2.8) bounds on the average value of $n(T)^2$ for N-key random 2-3 trees.

2.3 Higher Order Analysis

The method used in the previous two subsections obviously can be generalized to obtain better approximations of $\bar{n}(N)$. By computing the average number of nodes in the lowest k levels, we can determine $\bar{n}(N)$ to an accuracy of $1/2(2^k-1)\times 100\,\%$, because at most $1/2^k$ of the keys are in nodes above the k lowest levels.

In practice, the above procedure is difficult to carry out for $k\geqq 3$. If $F(k)$ is the number of different types of trees of height k, the solution of the problem involves the manipulation of an $F(k)\times F(k)$ matrix. A crude estimate shows that $F(3)\approx 200$ and $F(4)\approx 10^6$, making it a complicated calculation even for $k=3$ and virtually impossible for $k\geqq 4$.

3. An Analysis of *B*-Trees

3.1 Introduction

A natural extension of 2–3 trees is the idea of "*B*-trees" [2] [5, p. 473]. A *B-tree of order m* is a tree in which the number of keys contained in any internal node other than the root is no greater than $m-1$ and no less than $\lceil m/2\rceil - 1$; the root contains no more than $m-1$ keys. Thus, 2–3 trees are just *B*-trees of order 3. To add a key to a node, we insert the new key into the other keys and check if the node now contains more than $m-1$ keys. If the answer is no, the insertion has been completed. Otherwise, we split the node into 2 nodes, one of which contains the smallest $\lceil m/2\rceil - 1$ keys and the other the $m - \lceil m/2\rceil$ largest keys; the one remaining key is then inserted into the parent node. *Random B-trees* are defined in exactly the same way as random 2–3 trees are defined.

We shall study $\bar{n}_m(N)$, the average number of nodes in the *B*-trees of order m resulting from N random insertions. An obvious bound was given in [5, p. 476].

$$\frac{1}{m-1} \leqq \frac{\bar{n}_m(N)}{N} \leqq \frac{1}{\lceil m/2\rceil - 1} + \frac{1}{N}. \tag{14}$$

In this section we shall consider the nodes at the lowest level, and do an analysis similar to the first order analysis done in Section 2.1. As we shall see, this analysis yields better results than the corresponding analysis for 2–3 trees, because a greater proportion of keys in a *B*-tree are stored in the lowest internal nodes as m becomes larger.

Define the following functions:

$$H(N) = \sum_{k=1}^{N} \frac{1}{k} \quad \text{for } N \geqq 1,$$

$$r(m) = \begin{cases} \dfrac{1}{m+1}\,(H(m) - H(m/2))^{-1}, & \text{if } m \text{ is even,} \\ \dfrac{1}{m+1}\,(H(m+1) - H((m+1)/2))^{-1}, & \text{if } m \text{ is odd.} \end{cases}$$

It is well known [4, p. 74] that $H(m) \sim \ln m + 0.58 + \dfrac{1}{2m} + \cdots$. A simple computation shows that $r(m) = \dfrac{1}{m}\,\dfrac{1}{\ln 2} + O\left(\dfrac{1}{m^2}\right)$. Our new bounds on $\bar{n}_m(N)$ are given below:

Theorem 3.1. For any $\varepsilon > 0$ and fixed m,

$$\left(1 + \frac{1}{m-1}\right) r(m) - \varepsilon \leqq \frac{\bar{n}_m(N)}{N} \leqq \left(1 + \frac{1}{\lceil m/2\rceil - 1}\right) r(m) + \varepsilon$$

when N is sufficiently large.

Corollary. For any $\varepsilon > 0$ and fixed m, $\left|\dfrac{\bar{n}_m(N)}{N} - \dfrac{1}{m}\,\dfrac{1}{\ln 2}\right| < \dfrac{C}{m^2} + \varepsilon$ for all sufficiently large N, where C is a constant independent of m and N.

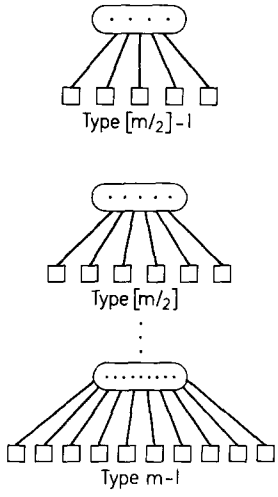

Fig. 5. The height 1 B-trees of order m consist of $m-\lceil m/2\rceil+1$ types (shown for $m=10$)

The Corollary follows from Theorem 3.1 and the approximation of $r(m)$ given earlier. If all the nodes in a B-tree of order m contain $m-1$ keys, there would be $N/(m-1)$ nodes. The ratio $N/((m-1)\bar{n}_m(N))$ can therefore be viewed as *storage utilization* [4]. Our corollary to Theorem 3.1 shows that, as N becomes large, the storage utilization is essentially $\ln 2 \sim 0.69$ for fixed large m [cf. Eq. (14)].

3.2 Proof of Theorem 3.1

We will first introduce some notations. Note that there are $m-\lceil m/2\rceil+1$ types of B-trees of order m and height 1. As shown in Figure 5, a type i B-tree contains i keys in its node for $i=\lceil m/2\rceil-1, \lceil m/2\rceil, \ldots, m-1$. A B-tree of order m is said to be *of class* $(y_{\lceil m/2\rceil-1}, y_{\lceil m/2\rceil}, \ldots, y_{m-1})$ if at the lowest level there are y_i subtrees of type i for each i. Let $P_N(y_{\lceil m/2\rceil-1}, y_{\lceil m/2\rceil}, \ldots, y_{m-1})$ be the probability for an N-key random B-tree of order m to be of class $(y_{\lceil m/2\rceil-1}, \ldots, y_{m-1})$.

Definition 3.2.

$$A_i(N)=\sum_{y_j\text{'s}} y_i\, P_N(y_{\lceil m/2\rceil-1}, \ldots, y_{m-1}) \qquad i=\lceil m/2\rceil-1, \ldots, m-1.$$

For brevity, we have suppressed the dependence of $A_i(N)$ and P_N on m in our notations.

Lemma 3.3.

$$\left(1+\frac{1}{m-1}\right)\left(\sum_{i=\lceil m/2\rceil-1}^{m-1} A_i(N)\right)-\frac{1}{m-1}$$
$$\leqq \bar{n}_m(N) \leqq \left(1+\frac{1}{\lceil m/2\rceil-1}\right)\left(\sum_{i=\lceil m/2\rceil-1}^{m-1} A_i(N)\right)-\frac{2}{\lceil m/2\rceil-1}+1.$$

Proof. Similar to the proof of Lemma 2.5. The term $+1$ appearing on the right-hand side of the equation arises from the fact that the root may contain less than $\lceil m/2 \rceil - 1$ keys. □

The major effort to prove Theorem 3.1 is contained in the next lemma.

Lemma 3.4. Let

$$g(N) = \sum_{i=\lceil m/2 \rceil - 1}^{m-1} A_i(N)/(N+1).$$

Then for any $\varepsilon > 0$,

$$|g(N) - r(m)| < \varepsilon \quad \text{for all sufficiently large } N.$$

Proof. We shall assume $m = 2p$ to be an even number. The proof for odd m is similar.

Let T be an $(N-1)$-key B-tree of order $2p$. After a random insertion, T may become a B-tree of class $(y_{p-1}, \ldots, y_{i-1} - 1,\ y_i + 1, \ldots, y_{2p-1})$ with probability $i y_{i-1}/N$ (for each $i = p, \ldots, 2p-1$) or it may become a B-tree of class $(y_{p-1} + 1,\ y_p + 1,\ y_{p+1}, \ldots, y_{2p-1} - 1)$ with probability $2p\, y_{2p-1}/N$. It follows that

$$A_{p-1}(N) = A_{p-1}(N-1) + \frac{1}{N}(-p A_{p-1}(N-1) + 2p A_{2p-1}(N-1)),$$

$$A_p(N) = A_p(N-1) + \frac{1}{N}(p A_{p-1}(N-1) - (p+1) A_p(N-1) + 2p A_{2p-1}(N-1))$$

and

$$A_j(N) = A_j(N-1) + \frac{1}{N}(j A_{j-1}(N-1) - (j+1) A_j(N-1)) \quad \text{for } p+1 \leqq j \leqq 2p-1. \tag{15}$$

Denoting by $A(N)$ the $(p+1)$-component vector $(A_j(N))$, (15) can be written in matrix notation as

$$A(N) = \left(I + \frac{1}{N} B\right) A(N-1) \tag{16}$$

where I is the $(p+1) \times (p+1)$ identity matrix, and B is defined by

$$B = \begin{pmatrix} -p & & & & & 2p \\ p & -(p+1) & & & & 2p \\ & p+1 & -(p+2) & & & 0 \\ & & p+2 & -(p+3) & \cdots & 0 \\ & & & \ddots & \ddots & \vdots \\ & & & & 2p-1 & -2p \end{pmatrix} \tag{17}$$

with zeroes elsewhere.

To solve (16) for $A(N)$, we define a $(p+1)$-component vector $a(N)=(a_i(N))$ by

$$a_i(N)=A_i(N)/(N+1). \tag{18}$$

Equations (16) and (18) lead to the following recurrence relation

$$a(N)=\left(I+\frac{1}{N+1}(B-I)\right)a(N-1). \tag{19}$$

The characteristic polynomial $q(\lambda)$ of $B-I$ is computed to be

$$q(\lambda)=(-1)^{p+1}(\lambda+2p+1)\left(\prod_{j=1}^{p}(\lambda+p+j)-\prod_{j=1}^{p}(p+j)\right).$$

We need the following lemma, whose proof can be found in Knuth [5, pp. 679–680, answer to ex. 10]:

Lemma 3.5. Let k be a positive integer, then the polynomial $g(\lambda)=\prod_{j=0}^{l}(\lambda+k+j)-\prod_{j=0}^{l}(k+j)$ has only simple roots. Furthermore, the real parts of all the roots except the root $\lambda=0$ are negative.

Using Lemma 3.5 it is easy to see that the roots of $q(\lambda)=0$ satisfy the following conditions:

(i) All roots are simple roots.
(ii) $\lambda=0$ is a root.
(iii) The real parts of all roots except $\lambda=0$ are negative.

Therefore, according to Lemma 2.10, there exists a vector

$$u=(u_{p-1}, u_p, \ldots, u_{2p-1})^T$$

such that

$$\text{(i)}\quad (B-I)u=0, \tag{21}$$

$$\text{(ii)}\quad |a_i(N)-u_i|<C_m N^{-\varepsilon_m} \quad \text{for } p-1\leqq i\leqq 2p-1 \tag{22}$$

where C_m, ε_m are positive constants.

Relation (22) implies

$$\left|\sum_{i=p-1}^{2p-1}A_i(N)/(N+1)-\sum_{i=p-1}^{2p-1}u_i\right|\leqq C'_m N^{-\varepsilon_m}. \tag{23}$$

Now, to determine the u_i's, we note that the following equation can be proved easily [cf. the derivation of Eq. (12)]:

$$\sum_{i=p-1}^{2p-1}(i+1)u_i=1. \tag{24}$$

Solving (21) and (24), we obtain

$$u_{p-1}=\frac{1}{p+1}\,\frac{1}{2p+1}\,(H(2p)-H(p))^{-1}$$

$$u_i=\frac{1}{i+1}\,\frac{1}{i+2}\,(H(2p)-H(p))^{-1} \qquad p\leqq i\leqq 2p-1. \tag{25}$$

Therefore,

$$\sum_{i=p-1}^{2p-1} u_i=\frac{1}{2p+1}\,(H(2p)-H(p))^{-1}=r(m). \tag{26}$$

Finally, substituting (26) into (23), the lemma is obtained. □

Proof of Theorem 3.1. It is a direct consequence of Lemma 3.3 and 3.4. □

4. Concluding Remarks

We have derived bounds on the average number of nodes in an N-key random B-tree. One interesting result is that the asymptotic storage utilization is approximately $\ln 2\approx 69\,\%$ for B-trees of high orders. This seems to agree well with one set of experimental data ($m=121$, $N=5000$, storage utilization $=67\,\%$, see [5, p. 479]).

Many problems about 2–3 trees remain to be investigated. What is the average number of splitting on the N-th random insertion [3]? How to analyze 2–3 trees when deletions are present? It appears that very different methods would be required to answer these questions satisfactorily.

Acknowledgments. I wish to thank the referees for helpful suggestions. The present analysis of Section 2.2, based on 7 types of height 2 trees, was recommended by a referee in preference to a bulkier, but perhaps more straightforward analysis using 9 types of trees in an earlier version [6] of this paper.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The design and analysis of computer algorithms. Reading (Mass.): Addison-Wesley 1974
2. Bayer, R., McCreight, E.: Organization and maintenance of large ordered indexes. Acta Informat. 1, 173–189 (1972)
3. Chvatal, V., Klarner, D.A., Knuth, D.E.: Selected combinatorial research problems. Computer Science Dept., Stanford University, Problem 37, STAN-CS-72-292, 1972
4. Knuth, D.E.: The art of computer programming, Vol. 1, Fundamental algorithms. Reading (Mass.): Addison-Wesley 1968
5. Knuth, D.E.: The art of computer programming, Vol. 3, Sorting and searching. Reading (Mass.): Addison-Wesley 1973
6. Yao, A.C.: On random 3–2 trees. Department of Computer Science, University of Illinois, Technical Report (74-679), October 1974

Received March 18, 1976