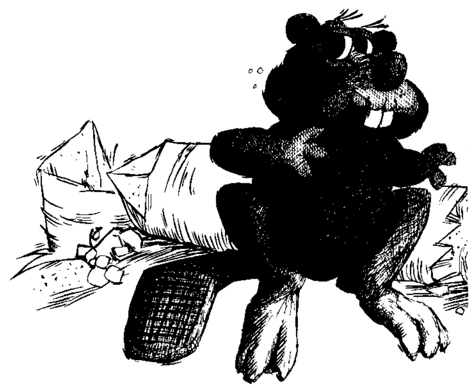
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF WATERLOO COMPUTER SCIENCE DEPARTMENT



The Expected Behaviour of B^+-Trees

Ricardo A. Baeza-Yates

Research Report
CS-86-68

December 1986

Expected Behaviour of B^+-Trees

Ricardo A. Baeza-Yates †

Computer Science Department
University of Chile
Blanco Encalada 2120
Santiago, CHILE

ABSTRACT

Fringe analysis is used to study the behaviour of B^+-trees (B-trees where all the records are stored in the leaves) under random insertions. We obtain bounds for the expected memory utilization of trees built using the usual insertion algorithm, the B^* overflow handling technique, and other techniques derived from the latter. Several other performance measures are also derived, such as bounds for the number of index nodes, the expected height, the expected insertion cost in the lowest level and the probabilities of 0, 1 and 2 or more splits per insertion. Special emphasis is place on 2-3 trees. A technique for concurrency in B^+-trees is also analyzed.

1. Introduction

Search trees, in particular balanced trees, (e.g. B-trees, 2-3 trees, height balanced trees) are efficient structures for storing and retrieving information. They are particularly suitable when a combination of requirements is considered, such as efficient access, ease of insertion and deletion of keys, memory utilization, and use of memory in two levels (main and secondary). Furthermore, they allow sequential access in key sequence.

The worst-case behaviour of m-ary search trees in their different forms (for instance B-trees, [4]) is well known [15]. However, few analytic results about their expected behaviour are available, even fewer if deletions are allowed. For the case of trees built by successive random insertions, Yao [22] introduced an analysis technique known as *Fringe Analysis,* which he used to find bounds for the expected number of nodes in B-trees. Such an analysis can be regarded as the first step in the study of the expected case. For the analysis in this paper, we also assume that the trees have been built by successive random insertions. This means that at each step there is an equal probability of inserting a new element in any of the $N+1$ intervals, where N is the number of keys in the tree. Fringe analysis only considers the lower part of the tree. Since part of the tree is not analyzed, this approach gives exact results only for those performance measures that depends only on the fringe, and bounds for other performance measures.

Yao's results were slightly extended by Brown [7], improved by Eisenbarth et al. [11], and by Baeza-Yates and Poblete [3]. In addition, Ziviani et al. [23] analyzed symmetric binary B-trees, while Culik et al. [10] introduced dense multiway trees, and studied a variant of these by using fringe analysis. An independent work [19] similar to Yao's paper also analyzes the transient behaviour of B^+-trees.

† Author's current address: Department of Computer Science, University of Waterloo, Waterloo, Ontario, CANADA N2L 3G1

By using a different technique, Kuspert [16] carried out the first theoretical study of B$^+$-trees using a generalized overflow technique, somewhat similar to the one used here.

This paper obtains a number of new results for B$^+$-trees built with or without overflow techniques: exact values for the expected number of buckets, the expected insertion cost at the bucket level, the probability of 0, 1 and 2 or more splits, and bounds in the expected storage utilization, the expected number of splits and the expected number of nodes in the index. Many of these results are new also for standard B-trees.

2. Basic definitions

We will define a B-tree of order m ($m \geq 1$) as follows:

(i) The root has between 2 and $2m + 1$ descendants.

(ii) Each internal node, except the root, has between $m + 1$ and $2m + 1$ descendants (hence the node contains from m to $2m$ keys).

(iii) All the leaves are at the same level.

The case m=1 defines 2-3 trees. Several important variants of B-trees have been proposed.

B$^+$-trees:

In a B$^+$-tree, all the keys are stored in the leaves. We use the notation introduced by Comer [9] to distinguish between B$^+$-trees and B*-trees. The keys can be easily scanned sequentially. The internal nodes contain information directing the access to the leaves (index). We will use the term *bucket* for external nodes and the term *index node* for internal nodes. Figure 1.1 shows a B-tree of this type.

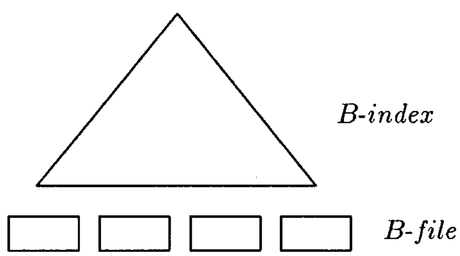


Figure 1.1. General B$^+$-tree.

B*-trees:

A B*-tree [16] is a B$^+$-tree with the following modification to the insertion algorithm: when overflow occurs, before carrying out a split, we inspect its neighbouring brothers. If are not both full, we reorganize two buckets so that a split is avoided; otherwise we take one brother and together with the original one we turn them into 3 buckets so that each one contains 2/3 of the keys. If we let b be the maximum number of keys that a bucket holds, then we have the process in Figure 1.2. In this way the minimum memory utilization is 2/3.

Figure 1.2. B* overflow technique.

Prefix B^+-trees:

Prefixes (separators) of the keys contained in the buckets are stored in the index nodes [6]. This saves space in the internal nodes, and we can accommodate more prefixes in an index node. Hence the order grows and, potentially, decreases the access time (height of the tree). If prefix setting is done only in the lowest level, the tree is said to be a simple prefix B^+-tree.

Figure 1.3 shows a prefix B^+-tree of order 1 with $b = 3$.

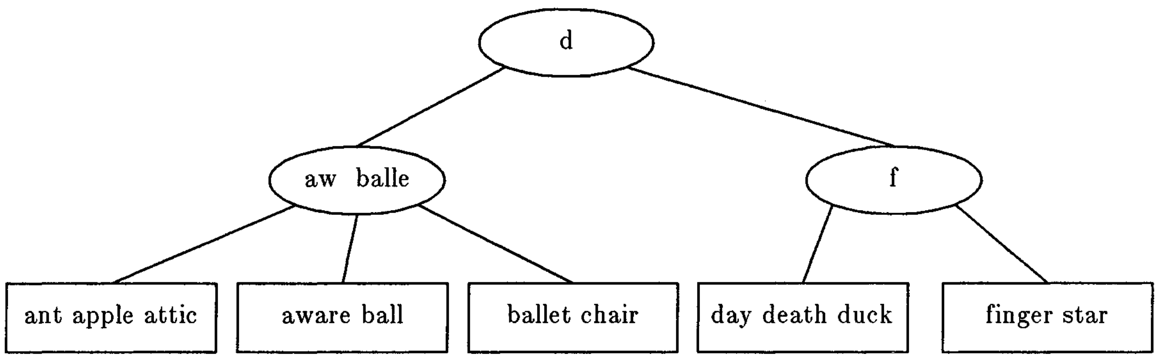


Figure 1.3. A Prefix B^+-tree.

3. Fringe Analysis

Fringe analysis has been formalized by Eisenbarth et al. [11], and the results from that paper are summarized in this section.

Let C be a finite *collection* $\{ T_1, \ldots, T_n \}$ of trees. Each tree $T_i \in C$ is said to be a type. The *fringe* of a tree is defined as a disjoint set of one or more subtrees isomorphic to elements in C. The fringe of a tree is hence not unique but generally includes all the subtrees fulfilling the above condition.

Example 3.1: Let C be the collection of all possible non-empty 2-3 trees of height 1. That is,

where each point represents a key. Using the above class, a fringe in a 2-3 tree is shown in Figure 3.1. The height of the fringe in a 2-3 tree is defined by the height of the elements in C.

The composition of a fringe can be described in several ways. Yao [22] did it in terms of the expected number of trees of each type in it. A more suitable description consists of using the probability of finding a leaf belonging to a type in the tree fringe [11]. Thus, if we have m types the above probability is expressed for the i-th type in an N-key tree as:

$$p_i(N) = \frac{\text{Expected number of leaves in type } i \text{ subtrees}}{N+1}, \quad i=1, \cdots, m \tag{3.1}$$

As an illustration of the method, example 3.1 produces the following transitions when an insertion is performed in each type:

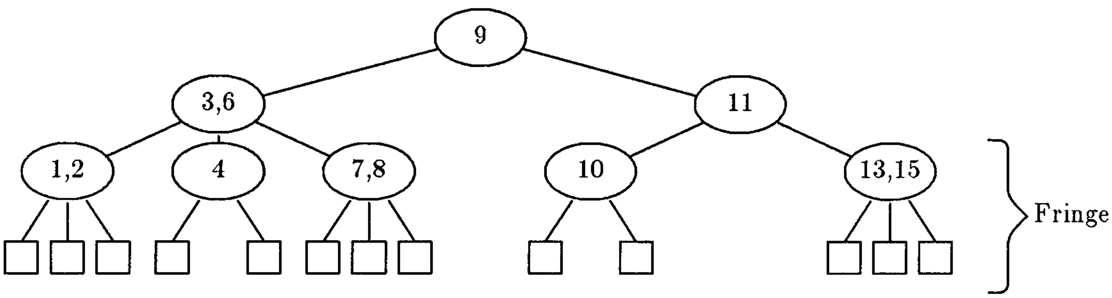


Figure 3.1: A 2-3 tree and the fringe defined by the set in Example 3.1.

$$Pr\{\ Type\ 1 \Rightarrow Type\ 2\ \} = p_1(N)$$
$$Pr\{\ Type\ 2 \Rightarrow Type\ 1 + Type\ 1\ \} = p_2(N)$$

Let $X_i(N)$ be the expected number of leaves belonging to type i subtrees after the N-th insertion. Since the set is closed, the numbers of leaves depends only on what has previously happened. Hence $X_i(N), i=1,2$ are random variables making up a Markov Chain, and so the process is a Markov process [11]. Therefore for type 1

$X_1(N)$	=	$X_1(N-1)$	−	$2p_1(N-1)$	+	$4p_2(N-1)$
Expected number of leaves after the insertion		Expected number of leaves before the insertion		Expected number of leaves lost		Expected number of leaves gained

Likewise for type 2

$$X_2(N) = X_2(N-1) - 3p_2(N-1) + 3p_1(N-1)$$

But $p_i(N) = \dfrac{X_i(N)}{N+1}$, for $i=1,2$; then in matrix notation we have

$$\vec{P}(N) = (\ I + \frac{1}{N+1} H\ )\ \vec{P}(N-1) \qquad (3.2)$$

where $\vec{P}(N) = (\ P_1(N), P_2(N)\ )^t$ and

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} -3 & 4 \\ 3 & -4 \end{bmatrix}$$

The matrix H is called the *transition matrix.*

The solution to a general fringe analysis problem is given by the following theorem.

Theorem 3.1. [11,p.134] Let H be the $m \times m$ transition matrix of a fringe analysis problem. Let $\lambda_1, \ldots, \lambda_m$ be H eigenvalues, with $\lambda_1 = 0 > \mathrm{Re}(\lambda_2) \geq \cdots \geq \mathrm{Re}(\lambda_m)$ [11, p. 134], and let $\vec{x}_1$ be the eigenvector corresponding to λ_1. Then for any vector $\vec{P}(0)$ there exist a constant c such that

$$|\,\vec{P}(N) - c\ \vec{x}_1\,| = O(\ N^{\mathrm{Re}(\lambda_2)}\ )$$

where $\vec{P}(N)$ is defined as in equation 3.2.

In other words, $\vec{P}(N)$ converges to the solution of $H\ \vec{x}_1 = 0$, where $\vec{x}_1$ is an m component vector not depending on N, when $N \rightarrow \infty$ Then

$$\vec{P}(N) = \alpha\, \vec{x}_1 \;+\; O(\, N^{\mathrm{Re}(\lambda_2)}\,) \tag{3.3}$$

where α is obtained from the condition $\alpha\, |\vec{x}_1| = 1$. Notice that $\mathrm{Re}(\lambda_2)$ is less than zero; in the following analysis this term will not be written explicitly. The numbers of subtrees of each type, $A_i(N)$, in the fringe can be obtained by rewriting equation 3.1 as

$$A_i(N) = \frac{p_i(N)\,(N+1)}{L_i} \tag{3.4}$$

where L_i is the number of leaves (intervals) of the type i.

4. Assumptions and Basic Formulas

In this section, we will state the assumptions made in the analysis and the basic formulas that will be needed later on.

We made the following additional assumptions:

(i) Keys are of fixed length. The keys may be of varying length if the order of the tree does not vary. For instance, the keys could have a maximum length. The analysis cannot be performed if prefixes are used in the internal nodes. This is because the intervals fail to be equally likely with prefixes of different lengths (i.e. short prefixes will imply higher probability of insertion into its associated interval).

(ii) The initial tree has only one key $x = -\infty$ which is less than any other key. This is done to prevent the leftmost bucket from having an extra interval, if we use the convention that all keys greater than or equal to a given one lie to the right. Hence, when there are N keys in the tree, there are N possible intervals where the next key to be inserted may fall. This alters the recurrence equation to

$$\vec{P}(N) = (\, I + \frac{1}{N}\, H\,)\, \vec{P}(N-1)$$

which also converges to the solution of equation 3.2. Notice that here the number of intervals for a bucket equals the number of keys stored in it.

We will study several cases of a general overflow technique for the last two levels. The insertion algorithm works as usual, except when an overflow occurs. In such a case it does not split the affected bucket but searches for free space among its brothers (if they exist) on both sides. A maximum number of bucket's brothers in each side may be examined. If available space is found, some keys are shifted, and a split is avoided. If not, a fixed number of buckets will form a new group one bucket larger. When forming the new group, keys are distributed as evenly as possible.

The technique is formalized as follows. Let ℓ be the maximum number of brothers to be examined on either side of a bucket in overflow when searching for available space ($\ell \geq 0$). Let k be the number of buckets grouped together for reorganization when performing a split ($k \geq 1$). We say that a B^+-tree is of type B_ℓ^k if it has been generated by random insertions using the overflow technique defined by ℓ and k. Clearly, the inequalities $\ell \leq 2m$ and $k \leq 2\ell + 1$ must hold. The parameter k is called the *split factor* by Kuspert [16], using $\ell = k-1$. For example, B_0^1 corresponds to a standard B^+-tree and B_1^2 to a B^*-tree.

The minimum memory utilization (used space over total space) is then

$$U_{\min} = \frac{k}{k+1}$$

Next, we introduce some properties of B^+-trees and the performance measures to be considered. Given the type i, let L_i be the number of keys at the bucket level, $nn(i)$ the number of index nodes, $nk(i)$ the number of keys in the index nodes, and p_i the probability of insertion into type i.

Definition 4.1. Given a B$^+$-tree of order m and buckets of size b keys, we shall define expected storage utilization as

$$\mathrm{E}\left(\frac{\textit{used space}}{\textit{total space}}\right)$$

We denote the expected bucket, index node, and total memory utilization when $N \to \infty$ by $\overline{U\ell}_b$, $\overline{Un}_m$ and $\overline{Ut}_m$, respectively.

Theorem 4.2. In a B$^+$-tree $\overline{Ut}_m$ converges to $\overline{U\ell}_b$ when $b \to \infty$, where b denotes the size of a bucket in keys.

Proof: We may write

$$Ut_m = \alpha\, Ul_b + (1 - \alpha)\, Un_m \ .$$

We shall now find an explicitly expression for α. We have

$$\alpha = \frac{\textit{Occupied space in the buckets}}{\textit{Total space}}$$

Let N_ℓ be the number of buckets and N_n the number of index nodes. Then,

$$\alpha = \frac{b\ N_\ell}{b\ N_\ell + 2m\ N_n} = \frac{1}{1 + \frac{2m}{b}\frac{N_n}{N_\ell}}.$$

In a p-ary tree we have that $N_\ell = (p-1)N_n + 1$. But the number of descendants of each index node is bounded by $m+1$ and $2m+1$, and therefore

$$\frac{1}{2m}(1 + \frac{1}{N_\ell}) \le \frac{N_n}{N_\ell} \le \frac{1}{m}(1 + \frac{1}{N_\ell})$$

N_ℓ is bounded by N/b and $N/bU_{\min}$ and hence

$$\frac{1}{1+\frac{2}{b}+\frac{2}{U_{\min}N}} \le \alpha \le \frac{1}{1+\frac{1}{b}+\frac{1}{N}}$$

which implies that $\lim_{N\to\infty, b\to\infty} \alpha = 1$, and therefore

$$\overline{Ut}_m \to \overline{U\ell}_b \ . \ \square$$

From the preceding theorem and because $\overline{Un}_m$ is bounded by $\frac{1}{2}$ and 1 we have

Corollary 4.3. For finite b, $\overline{Ut}_m$ is bounded by:

$$\frac{b\ \overline{U\ell}_b + 1}{b+2} \le \overline{Ut}_m \le \frac{b\ \overline{U\ell}_b + 1}{b+1} \ .$$

The bounds for $\overline{Un}_m$ and $\overline{Ut}_m$ can be improved by finding closer-fitting bounds for α, particularly for the lower one.

Theorem 4.4.

$$\overline{U\ell}_b \ge \frac{1}{\sum_i \frac{t_i\ p_i}{L_i}}$$

where t_i is the maximum number of keys that buckets of type i can hold.

Proof: $\overline{U\ell}_b$ is the product of the expected occupied memory (denoted by $\overline{M}_o$) and the expected reciprocal of the total memory used in the buckets. We denote the total expected memory by $\overline{M}_t$. Since in general $E(\frac{1}{X}) \neq \frac{1}{E(X)}$ we use Kantorovich inequality [8]

$$1 \leq E(X)\, E(\frac{1}{X}) \leq 1 + \frac{(X_{\max} - X_{\min})^2}{4X_{\min}X_{\max}}$$

In our case $X_{\min} = (N+1)/b$ and $X_{\max} = (N+1)/bU_{\min}$. Hence we have

$$\frac{\overline{M}_o}{\overline{M}_t} \leq \overline{U\ell}_b \leq (1 + \frac{(1-U_{\min})^2}{4U_{\min}})\, \frac{\overline{M}_o}{\overline{M}_t}$$

We are interested in the minimum value, so we will use

$$\overline{U\ell}_b \geq \frac{\overline{M}_o}{\overline{M}_t} = \frac{\sum_i L_i\, A_i(N)}{\sum_i t_i\, A_i(N)}$$

Substituting $p_i = \dfrac{A_i(N)\, L_i}{N+1}$, gives

$$\overline{U\ell}_b \geq \frac{(N+1)\sum_i p_i}{(N+1)\sum_i t_i \frac{p_i}{L_i}} = \frac{1}{\sum_i \frac{t_i\, p_i}{L_i}} \quad . \ \square$$

Lemma 4.5. [11,p.163] Let $\overline{n\ell}_m$ be the number of nodes at level ℓ in a B-tree of order m containing N keys. Then, the number of nodes above this level, $na\ell_m$, is bounded by

$$\frac{\overline{n\ell}_m - 1}{2m} \leq na\ell_m \leq \frac{\overline{n\ell}_m - 1}{m}$$

Proof: Consider level ℓ as being the $\overline{n\ell}_m$ leaves of a B-tree with $\overline{n\ell}_m - 1$ keys. (Each leaf represents a node.). The *minimum* and the *maximum* number of nodes above level ℓ is obtained when each node above the level ℓ contains $2m$ and m keys respectively. (That is $2m \times na\ell_m = \overline{n\ell}_m - 1$ and $m \times na\ell_m = \overline{n\ell}_m - 1$ respectively.). □

Lemma 4.5. and equation 3.4 lead to the following theorem:

Theorem 4.6. The expected number of index nodes in a random B⁺-tree of order m and N keys is bounded by

$$\sum_i \left(nn(i) + \frac{1}{2m} \right) \frac{p_i}{L_i}(N+1) - \frac{1}{2m} \leq \bar{n}_m(N) \leq \sum_i \left(nn(i) + \frac{1}{m} \right) \frac{p_i}{L_i}(N+1) - \frac{1}{m}$$

Lemma 4.7. [3] Let fh denote the height of the fringe (buckets included). The expected probability that fh or more splits occur due to one insertion is

$$Pr\{fh \text{ or more splits}\} = \sum_{i,j} h_{ji} \frac{p_i}{L_j}$$

where h_{ji} is the conditional probability that an insertion in type i splits the root producing a type j. Analogously, the expected probability that $fh-1$ splits occur due to one insertion is

$$Pr\{fh-1 \text{ splits}\} = \sum_{i,j} h'_{ji} \frac{p_i}{L_j}$$

where h'_{ji} is the conditional probability that an insertion in type i increments in one the number of elements in the root producing a type j.

Proof: If a root splits, we know that fh splits must have occurred. Given our unawareness of events above the root, this number may be larger. Analogously if a root of a subtree in the fringe becomes larger by one key, then we know that a split has occurred at each level below it [3]. □

Note that $Pr\{1\ split\}$ is the probability of one split at the bucket level. The next split will occur at the index nodes. It is known that the expected number of splits at any level, in one insertion, is

$$Pr\{split\ at\ level\ j\} \leq \frac{1}{m}$$

and that for large values of N and m the expected number of splits in a B-tree approaches the probability of one split [21]. The last equation gives us an upper bound for the expected number of splits in a B-tree of order m ($m>1$)

$$E[splits] \leq \sum_{j=1}^{\infty} \frac{j}{m^j} = \frac{m}{(m-1)^2} = \frac{1}{m} + O(m^{-2})\ .$$

In a B$^+$-tree, the only difference is that

$$Pr\{split\ at\ the\ bucket\ level\} \leq \frac{2}{b}$$

and hence

$$E[splits] \leq \frac{2}{b} \sum_{j=1}^{\infty} \frac{j}{m^{j-1}} = \frac{2m^2}{b(m-1)^2} = \frac{2}{b} + O(\frac{1}{bm})\ .$$

This can be further improved using the exact values of $Pr\{j\ splits\}$, for j from 0 to $fh-1$, and the value of $Pr\{fh\ or\ more\ splits\}$. That is, for $m>1$

$$E[splits] \leq \sum_{j=1}^{fh-1} j\ Pr\{j\ splits\} + Pr\{fh\ or\ more\ splits\} \sum_{j=fh}^{\infty} \frac{j}{m^{(j-fh)}}\ ;$$

for $fh=2$, and for $m>1$

$$E[splits] \leq Pr\{1\ split\ \} + Pr\{2\ or\ more\ splits\} \frac{m(2m-1)}{(m-1)^2}$$

$$\leq Pr\{1\ split\ \} + 2Pr\{2\ or\ more\ splits\} + O(1/m)$$

Let $c(m) = \frac{m(2m-1)}{(m-1)^2}$. Table 4.1 shows how $c(m)$ converges to 2.

m	2	3	4	5	6	7	8
$c(m)$	6	3.75	3.11	2.81	2.64	2.53	2.45

Table 4.1. Values for $c(m)$.

Let $\bar{I}_{acc}$ be the expected number of accesses at the buckets level for one insertion. This cost depends on $Pr\{1\ split\ or\ more\}$. At the buckets level two accesses to secondary memory are needed. If a split is required at least four accesses are needed: one read, two writes for the new buckets and one write to update the index, i.e.

$$\bar{I}_{acc} = 2 + 2Pr\{1\ split\ or\ more\}$$

For B$^+$-trees with overflow techniques the number of accesses in one insertion will be between 2 and $3\ell + 2$ if splitting does not occur (read $2\ell+1$ and write $\ell+1$ buckets). If a split occurs, $2\ell + k + 3$ accesses are performed (read $2\ell+1$ and write $k+1$ buckets plus one access to update the index). Therefore

$$2 + (2\ell + k + 1)Pr\{1 \textit{ split or more}\} \leq \bar{I}_{acc} \leq 3\ell + 2 + (k - \ell + 1)Pr\{1 \textit{ split or more}\}$$

Hence, overflow techniques improve storage utilization but require more accesses. It is possible, but laborious, to compute the expected number of accesses when a split does not occur. Also, if we know the probability of a split at the index level, we can compute additional accesses in the index. However, this cost is insignificant compared with the cost at the buckets level.

Lemma 4.8. The expected number of keys of the index nodes in the fringe of an order m B$^+$-tree with N keys is

$$\bar{f}_m(N) = \sum_i nk(i) \frac{p_i}{L_i}(N+1)$$

and the expected total number of keys in the index is

$$\overline{nk}_m(N) = \bar{f}_m(N) + \overline{n\ell}_m(N) - 1 = \sum_i (nk(i) + 1) \frac{p_i}{L_i} (N+1) - 1$$

where $\overline{n\ell}_m(N)$ denotes the expected number of nodes at the level of the roots of subtrees in the fringe.

Proof: The total number of keys of the index nodes in the fringe is

$$\sum_i nk(i) A_i(N).$$

The total number of keys in the index will be equal to the number of keys in the fringe plus the number of those above it, namely $\overline{n\ell}_m(N) - 1$ (as in Lemma 4.5).□

Lemma 4.9. The expected height, $\bar{h}_m(N)$, (buckets included) of a B$^+$-tree of order m and N keys is bounded by

$$\bar{h}_m(N) \leq \log_{m+1}(N+1) + \log_{m+1}\left(\sum_i \frac{p_i}{2L_i} \right) + fh + 1$$

Note that the second term is always negative.

Proof: Let $na\ell_m$ be the number of keys above level ℓ of the fringe's roots of height fh. Then we have (considering a binary root)

$$\bar{h}_m(N) \leq \left\lfloor \log_{m+1}\left(\frac{na\ell_m + 1}{2}\right) \right\rfloor + fh + 1$$

Considering the expected value of the right hand side of the above inequality then

$$\bar{h}_m(N) \leq \mathrm{E}\left[\left\lfloor \log_{m+1}\left(\frac{na\ell_m + 1}{2}\right) \right\rfloor + fh + 1 \right] \leq E\left[\log_{m+1}\left(\frac{na\ell_m + 1}{2}\right)\right] + fh + 1$$

Using Jensen's inequality [13, p.180] we obtain

$$\bar{h}_m(N) \leq \log_{m+1}\left(\frac{\mathrm{E}[na\ell_m + 1]}{2}\right) + fh + 1$$

But $\mathrm{E}[na\ell_m + 1] = \overline{n\ell}_m$. Substituting $\overline{n\ell}_m$ in the above equation we obtain the result.□

The above bound should be compared with the one given by

$$\bar{h}_m(N) \leq \left\lceil \log_{m+1}\left(\frac{1}{2} \left\lceil \frac{N+1}{b\, U_{\min}} \right\rceil \right) \right\rceil + 2$$

where $\left\lceil \frac{N+1}{b\, U_{\min}} \right\rceil$ is the highest number of buckets that a B$^+$-tree can have. The added constant 2 appears considering that the level of the buckets is counted in and that the root of the tree could be binary.

Theorem 4.10. Memory utilization in the index nodes is bounded by

$$\frac{\lceil \overline{nk}_m(N)/2m \rceil}{nsup(N)} \le \overline{Un}_m(N) \le \frac{9\lceil \overline{nk}_m(N)/2m \rceil}{8ninf(N)}$$

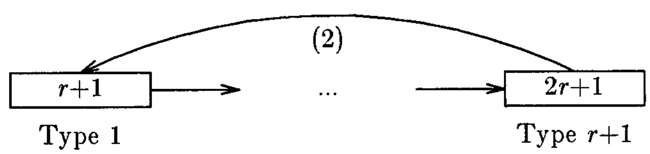
where $ninf(N)$ and $nsup(N)$ are the lower and upper bounds of theorem 4.6 respectively. The factor $9/8$ appears because we are using Kantorovich inequality with $U_{\min} = 1/2$.

Proof: The memory utilization in a B-tree of order m and N keys is defined as $\frac{\lceil N/2m \rceil}{\overline{n}_m(N)}$ [11,21], where $\lceil N/2m \rceil$ is the smallest number of nodes (when all the nodes contain $2m$ keys). In a B$^+$-tree there are $\overline{nk}_m(N)$ keys in the index and $\overline{n}_m(N)$ is bounded by theorem 4.6. □

5. The simple B$^+$ case.

The first level analysis is independent of the order of the tree, because it is performed on the bucket's level. Here the transitions depend on whether the bucket size, b, is odd or even. However, the asymptotic results are the same for both cases.

Let $b = 2r+1$ be odd. Then the first type has $r+1$ keys in one bucket and the $(r+1)$-th type has a full bucket. After an insertion type i transforms into a type $i+1$ and type $r+1$ transforms into 2 of type 1 (split).



Hence the transition matrix becomes

$$H = \begin{bmatrix} -(r+2) & & & 2r+2 \\ r+2 & -(r+3) & & \\ & \cdot & \cdot & \\ & & \cdot\ -(2r+1) & \\ & & 2r+1 & -(2r+2) \end{bmatrix}$$

which gives the solution $p_i = \frac{1}{(r+i+1)} \frac{1}{(H_{r+2} - H_{r+1})}$. H_n are the harmonic numbers of first order [14, p.73] defined by

$$H_n = \sum_{i=1}^{n} \frac{1}{i} = \ln n + \gamma + \frac{1}{2n} + O(n^{-2})$$

where γ is the Euler's constant (0.577..). It is important to note that the probability distribution is proportional to the reciprocal of the number of keys in the bucket. This distribution will be similar in the other cases. Hence the assumption of a uniform distribution made in Leung [17] is incorrect. This distribution is different from the distribution resulting from the assumption that all possible configurations of B-trees are equally likely [12]. Hence we find

$$Pr\{\ 0\ splits\ \} = 1 - \frac{1}{(b+1)\,(H_{b+1} - H_{(b+1)/2})} = 1 - \frac{1}{b\ln 2} + O(b^{-2})$$

$$Pr\{\ 1\ or\ more\ splits\ \} = \frac{1}{(b+1)\,(H_{b+1} - H_{(b+1)/2})} = \frac{1}{b\ln 2} + O(b^{-2})$$

$$\overline{nk}_m(N) = \frac{1}{(b+1)\,(H_{b+1} - H_{(b+1)/2})}\,(N+1) = (\frac{1}{b\ln 2} + O(b^{-2})\,)\,(N+1)$$

$$\bar{I}_{acc} = 2 + \frac{2}{b \ln 2} + O(b^{-2})$$

$$\frac{1}{2m(b+1)} \frac{(N+1)}{(H_{b+1} - H_{(b+1)/2})} - \frac{1}{2m} \leq \bar{n}_m(N) \leq \frac{1}{m(b+1)} \frac{(N+1)}{(H_{b+1} - H_{(b+1)/2})} - \frac{1}{m}$$

$$\bar{h}_m(N) \leq \log_{m+1}(N+1) - \log_{m+1}(2(b+1)\,(H_{b+1} - H_{(b+1)/2})) + 2 \approx \log_{m+1}(\frac{N+1}{2\ln 2\; b}) + 2$$

$$\frac{1}{2} + O(\frac{1}{m}) \leq \overline{Un}_m(N) \leq 1 - O(\frac{1}{m})$$

$$\overline{U\ell}_b \geq \frac{b+1}{b}\,(H_{b+1} - H_{(b+1)/2}) = \ln 2 + O(\frac{1}{b})$$

When b is even the results are almost identical. The term $(H_b - H_{(b+1)/2})$ is replaced by $(H_b - H_{b/2})$, which has the same asymptotic value. We have

$$\lim_{b \to \infty} \overline{Ut}_m = \lim_{b \to \infty} \overline{U\ell}_b = \ln 2 \approx 0.6931$$

the same as in B-trees [11,19,21,22]. Table 5.1. shows the values of the measures that do not depend on m for a few different bucket sizes

b	1	2	3	4	5	6	7	...	∞
$\overline{U\ell}_b$	1	.750	.777	.729	.740	.719	.725	...	.6931
$\frac{\overline{nk}_m(N)}{(N+1)}$	0	.666	.429	.343	.270	.232	.197	...	0
$Pr\{$ 0 splits $\}$	0	.333	.571	.657	.730	.768	.803	...	1
$Pr\{$ 1 or more splits $\}$	1	.666	.429	.343	.270	.232	.197	...	0
$\bar{I}_{acc}$	4	3.33	2.86	2.69	2.54	2.46	2.39	...	2

Table 5.1. Measures of a simple B$^+$-tree depending on b.

Note that $\overline{U\ell}_b$ is non monotone and tends to ln 2 from above (see figure 8.1). Hence it is convenient to use b odd and not too large (for example, the size given by an I/O operation in secondary memory).

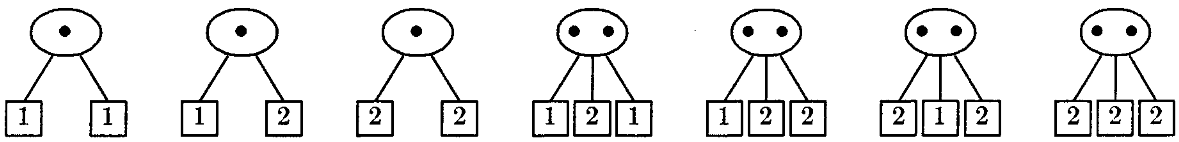
The total memory utilization is therefore bounded by

$$\overline{Ut}_m \geq \frac{(b+1)\,(H_{b+1} - H_{(b+1)/2}) + 1}{b+2}$$

For example, for a B$^+$-tree of order $m = 9$, with $b = 9$ and $N = 10000$, the expected height is less than or equal to 4 (the maximum is 5), the memory utilization exceeds 67.8%, and the probability that a split occurs due to an insertion is near 0.15.

The results above can be improved with a second order analysis. However, a general analysis cannot be formulated for arbitrary values of m and b since the number of types increases rapidly and transitions can only be formulated in terms of specific values of m and b. For that reason the analysis was done only for some small values of m and b. The case $m = 1$ (2-3$^+$ tree), $b = 2$ is shown in more detail.

The possible types are



Type 1 Type 2 Type 3 Type 4 Type 5 Type 6 Type 7

Each bucket indicates the number of keys that it contains. For a k-ary type in a second-order analysis, with $k = m+1, \cdots, 2m+1$, the notation $n_1, \ldots, n_k$ will denote the number of keys in bucket $1, \ldots, k$ of the type. In the preceding set, symmetry (e.g. type 12 behaves like type 21) and structure (type 121 is similar to type 112) was used. The associated transition matrix is

$$H = \begin{bmatrix} -3 & & & 1 & 2/5 & & \\ 3 & -4 & & 3/2 & 18/5 & & 3 \\ & 4/3 & -5 & & 4/5 & 24/5 & 4 \\ & 8/3 & & -5 & & & \\ & & 5/2 & 15/4 & -6 & & \\ & & 5/2 & 5/4 & & -6 & \\ & & & & 6/5 & 6/5 & -7 \end{bmatrix}$$

which gives the solution

$$\vec{P} = \{\ 0.07936,\ 0.35754,\ 0.14135,\ 0.19068,\ 0.11848,\ 0.07876,\ 0.03381\ \}$$

Hence we find

$$Pr\{\ 0\ splits\ \} = p_1 + \frac{p_3}{3} + \frac{p_4}{2} + \frac{1}{5}(p_6 + p_5) = \frac{1}{3}$$

$$Pr\{\ 1\ split\ \} = \frac{2}{3}p + p_3 = 0.3797$$

$$Pr\{\ 2\ or\ more\ splits\ \} = \frac{p_4}{2} + \frac{4}{5}(p_5 + p_6) + p_7 = 0.28695$$

$$\overline{nk}_1(N) = \frac{2}{3}(N+1)$$

$$\bar{h}_1(N) \leq \log_2(N+1) + 0.1989$$

$$0.43043\ (N+1) - \frac{1}{2} \leq \bar{n}_1(N) \leq 0.57391\ (N+1) - 1$$

$$0.58706 \leq \overline{Un}_1 \leq 0.87122\ ,\quad \overline{Ul}_2 \geq 0.75$$

$$\overline{Ut}_1 \geq 0.67173$$

The upper bound in the memory utilization is computed multiplying the lower bound by $9/8$ ($U_{\min} = 1/2$). Tables 5.2 to 5.5 show the results of selected variables for some values of m and b.

m	b	1	2	3	4	5	6	7	8
1	$Pr\{1\ splits\}$	.5714	.3797	.2465	.1975	.1554	.1335	.1130	.1007
	$Pr\{2\ or\ more\ splits\}$	.4286	.2870	.1821	.1454	.1149	.0982	.0840	.0744
2	$Pr\{1\ splits\}$	.7297	.4866	.3137	.2510	.1980			
	$Pr\{2\ or\ more\ splits\}$	.2703	.1800	.1149	.0919	.0720			
	$E[splits]$	2.3515	1.5667	1.0031	.8024	.0630			
3	$Pr\{1\ splits\}$	.8030	.5355	.3447					
	$Pr\{2\ or\ more\ splits\}$	.1970	.1312	.0838					
	$E[splits]$	1.5418	1.0275	.6590					
4	$Pr\{1\ splits\}$	.8451	.5635						
	$Pr\{2\ or\ more\ splits\}$	.1549	.1031						
	$E[splits]$	1.3270	.8843						
5	$Pr\{1\ splits\}$	.8724							
	$Pr\{2\ or\ more\ splits\}$	.1276							
	$E[splits]$	1.2310							

Table 5.2. Probability of 1 split, 2 or more splits and upper bound on $E[splits]$.

$m \setminus b$	1	2	3	4	5	6	7	8
1	.6429	.4304	.2731	.2180	.1723	.1472	.1261	.1117
	.8571	.5739	.3642	.2907	.2298	.1963	.1681	.1489
2	.3378	.2250	.1436	.1148	.0904			
	.4054	.2700	.1723	.1378	.1084			
3	.2298	.1531	.0978					
	.2627	.1749	.1118					
4	.1742	.1160						
	.1936	.1289						
5	.1403							
	.1531							

Table 5.3. Lower and upper bound on $\frac{\overline{n}_m(N)}{(N+1)}$.

$m \setminus b$	1	2	3	4	5	6	7	8
1	.5833	.5808	.5884	.5897	.5881	.5901	.5860	.5881
	.8750	.8712	.8827	.8846	.8821	.8851	.8790	.8821
2	.6167	.6171	.6219	.6221	.6232			
	.8325	.8331	.8395	.8398	.8413			
3	.6345	.6352	.6390					
	.8158	.8167	.8216					
4	.6456	.6463						
	.8070	.8079						
5	.6532							
	.8017							

Table 5.4. Lower and upper bound on $\overline{Un}_m$.

$m \setminus b$	1	2	3	4	5	6	7	8
1	.7368	.6717	.7093	.6877	.7015	.6909	.6779	.6918
2	.7629	.6905	.7234	.6985	.7116			
3	.7764	.6995	.7302					
4	.7847	.7048						
5	.7902							

Table 5.5. Lower bound on $\overline{Ut}_m$.

6. B$^+$-trees with Overflow Techniques

In this section we use a second-order analysis, which makes up the minimal fringe for studding the effect of our overflow techniques. Three cases, representatives of the generalized technique defined previously, will be studied, namely B_1^2, B_{2m}^2 and B_{2m}^{2m+1}.

6.1. The B_1^2 case.

Here, we study the simplest technique (B*-trees) for small values of m and b, since the transition matrix cannot easily be generalized. For the value of b, we use a multiple of 3 (see figure 1.2). The simplest case is for $m = 1$ and $b = 3$ and the types are: 22, 23, 33, 223, 232, 323, 233, and 333.

The associated transition matrix is:

$$H = \begin{bmatrix} -5 & & & & & & 3/2 & 4/3 \\ 5 & -6 & & & & & 15/8 & 20/3 \\ & 6 & -7 & & & & & 2 \\ & & 14/3 & -8 & & & & \\ & & 7/3 & & -8 & & & \\ & & & & 16/7 & -9 & & \\ & & & 8 & 40/7 & & -9 & \\ & & & & & 9 & 45/8 & -10 \end{bmatrix}$$

The solution is:

$$\vec{P} = (0.0714, 0.2250, 0.2240, 0.1307, 0.0653, 0.0332, 0.1411, 0.1092)$$

Hence we get

$$Pr\{0 \ \ splits\} = \frac{9863}{16068} = 0.6138$$

$$Pr\{1 \ \ split\} = p_3 = \frac{300}{1339} = 0.2240$$

$$Pr\{2 \ or \ more \ splits\} = \frac{3}{8}\, p_7 + p_8 = \frac{2605}{16068} = 0.1621$$

$$\bar{I}_{acc} = 2p_1 + \frac{16}{5}p_2 + 6p_3 + \frac{20}{7}p_4 + \frac{7}{2}p_5 + \frac{17}{4}p_6 + 8p_7 = 4.28$$

$$\overline{nk}_1(N) = 0.3862\ (N{+}1)$$

$$0.2432\ (N{+}1) - \frac{1}{2} \le \bar{n}_1\ (N)\ \le 0.3442\ (N{+}1) - 1$$

$$h_1(N) \le \log_2 (N{+}1) - 0.6248$$

$$0.5955\ \le \overline{Un}_1\ \le\ 0.8932$$

$$0.86317 \le \overline{Ul}_3 \le 0.89914$$

$$\overline{Ut}_1 \geq 0.7671$$

Here, the right side of the Kantorovich inequality is 25/24 because $U_{\min} = 2/3$. Tables 6.1.1 to 6.1.5 show the results of selected variables for some values of m and b.

m	b	3	6	9	12	15	18
1	*Pr{0 splits}*	.6138	.8023	.8671	.8999	.9197	.9330
	Pr{1 split}	.2240	.1160	.0784	.0591	.0476	.0398
	Pr{2 or more splits}	.1621	.0816	.0545	.0409	.0328	.0273
	$\bar{I}_{acc}$	3.93	2.99	2.66	2.50	2.40	2.34
2	*Pr{0 splits}*	.6199	.8053	.8689	.9011		
	Pr{1 split}	.2790	.1430	.0962	.0725		
	Pr{2 or more splits}	.1011	.0517	.0349	.0264		
	$\bar{E}[splits]$	.8856	.4532	.3056	.2309		
	$\bar{I}_{acc}$	3.90	2.97	2.66	2.49		
3	*Pr{0 splits}*	.6222					
	Pr{1 split}	.3044					
	Pr{2 or more splits}	.0734					
	$\bar{E}[splits]$	.5797					
	$\bar{I}_{acc}$	3.89					
4	*Pr{0 splits}*	.6235					
	Pr{1 split}	.3188					
	Pr{2 or more splits}	.0576					
	$\bar{E}[splits]$	.4980					
	$\bar{I}_{acc}$	3.88					

Table 6.1.1. Probabilities of 0 splits, 1 split, and 2 or more splits, upper bound on $E[splits]$ and lower bound on $\bar{I}_{acc}$.

$m \setminus b$	3	6	9	12	15	18
1	.2432	.1225	.0818	.0614	.0491	.0409
	.3242	.1633	.1091	.0819	.0655	.0546
2	.1264	.0647	.0436	.0320		
	.1517	.0776	.0524	.0395		
3	.0856					
	.0979					
4	.0648					
	.0720					

Table 6.1.2. Lower and upper bounds on $\dfrac{\bar{n}_m(N)}{(N+1)}$.

6.2. The B^2_{2m} case

In this technique, we seek over all the brothers for free space. When there is no free space, we take two buckets to turn them into three, just as in the B* method.

The types are:

$$\underbrace{\frac{2}{3}b, \frac{2}{3}b+1, \cdots, b}_{m+1} \rightarrow \cdots \rightarrow \underbrace{\frac{2}{3}b, b, \cdots, b}_{m+1} \rightarrow \underbrace{b, b, \cdots, b}_{2m+1}$$

Hence, we have from $mb + b/3 + 1$ to $(2m + 1)b$ buckets, i.e. $(m+1)b - b/3$ types. For b=3, we have $2b/3 + 1 = b$, which modifies the transition matrix. However the formulae are also

$m \setminus b$	3	6	9	12	15	18
1	.5955 .8932	.6053 .9080	.6093 .9140	.6115 .9173	.6130 .9195	.6141 .9211
2	.6266 .8459	.6273 .8469	.6262 .8454	.6253 .8442		
3	.6434 .8272					
4	.6532 .8165					

Table 6.1.3. Lower and upper bounds on $\overline{Un}_m$

$m \setminus b$	3	6	9	12	15	18
1	.8632	.8431	.8359	.8322	.8300	.8286
2	.8770	.8559	.8473	.8425		
3	.8823					
4	.8854					

Table 6.1.4. Lower bounds on $\overline{U\ell}_b$

$m \setminus b$	3	6	9	12	15	18
1	.7671	.7918	.8009	.8058	.8088	.8108
2	.7900	.8079	.8140	.8170		
3	.8008					
4	.8070					

Table 6.1.5. Lower bound on $\overline{Ut}_m$

valid in this case.

Let $h_i = mb + b/3 + 1$, $h_m = mb + 2b/3$, and $h_f = (2m + 1)b$, then the transition matrix is:

$$H = \begin{bmatrix} -(h_i + 1) & & & & h_i \\ h_i + 1 & & & & \\ & & \cdot & & \\ & & -(h_m + 1) & & h_m \\ & & h_m + 1 & & \\ & & & \cdot & \\ & & & \cdot & -(h_f + 1) \end{bmatrix}$$

whose solution is (numbering the types starting from h_i instead from 1 in order to let the number of buckets of each type be more easily noticed):

$$p_{h_i} = \frac{(mb + b/3 + 2)^{-1}}{H_{mb+2b/3} - H_{mb+b/3+1} + \beta\,(H_{(2m+1)b+1} - H_{mb+2b/3})} ,$$

$$p_i = \frac{mb + b/3 + 2}{i+1}\; p_{h_i} \qquad i = h_i,\ldots,h_m - 1 \quad \text{and}$$

$$p_i = \frac{mb + b/3 + 2}{i+1}\; \beta\, p_{h_i} \qquad i = h_m,\ldots,h_f \quad ,$$

with

$$\beta = \frac{(2m+1)b+1}{mb+b/3+1} .$$

Thence

$$Pr\{0 \; splits\} = 1 - \frac{1}{b}\,(1 + \frac{1}{2m\ln 2}) + O(m^{-2}, b^{-2})$$

$$Pr\{1 \; split\} = \beta\,(mb + \frac{b}{3} + 2)\,(\Psi(2m+1+1/b) - \Psi(m+1+1/b))\,\frac{p_{h_i}}{b} = \frac{1}{b} + O(m^{-2}, b^{-2})$$

where $\Psi(z)$ is the function defined by [1, p.258]

$$\Psi(z) = \frac{d}{dz}\ln(\Gamma\,(z)) = \ln z - \frac{1}{2z} + O(z^{-2})$$

$$Pr\{2 \; or \; more \; splits\} = p_{h_f} = \frac{1}{2mb\ln 2} + O(m^{-2}, b^{-2})$$

$$\bar{I}_{acc} \geq 2 + \frac{4m}{b} + \frac{1}{b}(3 + \frac{2}{\ln 2}) + O(1/mb)$$

$$\overline{n\ell}_m(N) = \frac{N+1}{(mb+b/3+1)(H_{mb+2b/3} - H_{mb+b/3+1}) + ((2m+1)b+1)(H_{(2m+1)b+1} - H_{mb+2b/3})}$$

$$(1 + \frac{1}{2m})\overline{n\ell}_m(N) \leq \bar{n}_m(N) \leq (1 + \frac{1}{m})\overline{n\ell}_m(N)$$

$$\bar{n}_m = \frac{1}{2mb\ln 2} + O(m^{-2}, b^{-2})$$

$$\overline{nk}_m(N) = \frac{\beta(\Psi(2m+2+1/b) - \Psi(m+1+1/b))}{b\,(H_{mb+2b/3} - H_{mb+b/3+1} + \beta(H_{(2m+1)b+1} - H_{mb+2b/3}))}$$

$$\bar{h}_m(N) \leq \log_{m+1}(\frac{N+1}{4b\ln 2}) + 2 + O(\frac{1}{m})$$

$$\overline{Un}_m \geq \frac{((2m+1)b+1)(\Psi(2m+2+1/b) - \Psi(m+1+1/b))}{(2m+2)\,b}$$

$$\overline{Un}_m \geq \ln 2 + \frac{1}{m}\,(1 - \frac{\ln 2}{2}) + O(\frac{1}{mb})$$

$$\overline{U\ell}_b \geq \frac{H_{mb+2b/3} - H_{mb+b/3+1} + \beta(H_{(2m+1)b+1} - H_{mb+2b/3})}{\beta(\Psi(2m+2+1/b) - \Psi(m+1+1/b))}$$

$$\lim_{b\to\infty} \overline{U\ell}_b \geq \frac{(m+1/3)(\ln(m+2/3) - \ln(m+1/3)) + (2m+1)(\ln(2m+1) - \ln(m+2/3))}{(2m+1)(H_{2m+1} - H_m)}$$

$$\lim_{b\to\infty} \overline{U\ell}_b = \lim_{b\to\infty} \overline{Ut}_m = 1 - \frac{1}{2m\,\ln 2} + O(m^{-2})$$

$$\lim_{m\to\infty} \overline{U\ell}_b = 1$$

$$\lim_{m\to\infty} \overline{Ut}_m \geq 1 - \frac{1}{b}\,(\ln^{-1}2 - 1) + O(b^{-2})$$

For the same B$^+$-tree mentioned in section 4 (m=9, b=9) the memory utilization will exceed 92%. Again, as in section 6.1, the upper bound in $\overline{U\ell}_b$ is the 25/24 times the lower bound. Tables 6.2.1 to 6.2.5 show the values of selected measures for some values of b and m.

m	b	3	6	9	12	15
1	*Pr{0 splits}*	.6238	.8055	.8687	.9009	.9204
	Pr{1 split}	.2213	.1155	.0782	.0591	.0475
	Pr{2 or more splits}	.1549	.0790	.0530	.0399	.0320
	$\bar{I}_{acc}$	4.63	3.36	2.92	2.69	2.56
2	*Pr{0 splits}*	.6388	.8155	.8760	.9067	.9251
	Pr{1 split}	.2669	.1369	.0921	.0694	.0556
	Pr{2 or more splits}	.0943	.0477	.0319	.0240	.0192
	$\bar{E}[splits]$	.8327	.4231	.2835	.2134	.1708
	$\bar{I}_{acc}$	5.97	4.03	3.36	3.03	2.82
3	*Pr{0 splits}*	.6460	.8202	.8794	.9093	.9273
	Pr{1 split}	.2862	.1457	.0977	.0735	.0589
	Pr{2 or more splits}	.0677	.0341	.0228	.0171	.0137
	$\bar{E}[splits]$	.5400	.2796	.1832	.1376	.1103
	$\bar{I}_{acc}$	7.31	4.70	3.81	3.36	3.09
4	*Pr{0 splits}*	.6502	.8229	.8814	.9109	.9286
	Pr{1 split}	.2969	.1505	.1008	.0758	.0607
	Pr{2 or more splits}	.0529	.0266	.0178	.0133	.0107
	$\bar{E}[splits]$	.4615	.2333	.1562	.1172	.0940
	$\bar{I}_{acc}$	8.65	5.36	4.25	3.69	3.36
5	*Pr{0 splits}*	.6530	.8247	.8827	.9119	.9294
	Pr{1 split}	.3036	.1535	.1028	.0772	.0618
	Pr{2 or more splits}	.0433	.0218	.0145	.0109	.0087
	$\bar{E}[splits]$	.4254	.2148	.1436	.1079	.0863
	$\bar{I}_{acc}$	9.98	6.03	4.70	4.03	3.62

Table 6.2.1. Probabilities of 0 splits, 1 split, and 2 or more splits, upper bound on $E[splits]$ and lower bound on $\bar{I}_{acc}$.

$m \setminus b$	3	6	9	12	15
1	.2323	.1185	.0796	.0599	.0481
	.3098	.1580	.1061	.0799	.0641
2	.1179	.0596	.0399	.0300	.0240
	.1414	.0715	.0479	.0360	.0288
3	.0790	.0398	.0266	.0200	.0160
	.0903	.0455	.0304	.0228	.0183
4	.0595	.0299	.0200	.0150	.0120
	.0661	.0332	.0222	.0167	.0133
5	.0478	.0239	.0160	.0120	.0096
	.0520	.0261	.0174	.0131	.0105

Table 6.2.2. Lower and upper bounds on $\dfrac{\bar{n}_m(N)}{(N+1)}$.

6.3. The B_{2m}^{2m+1} case

This is the limiting case for the set of overflow techniques considered in this paper. Here, the free space is distributed among all the brothers. In fact, when there is overflow in the $2m+1$ buckets of the type $bb \cdots bb$ this turns into two subtrees with $m+1$ buckets of type $(b-a)(b-a) \cdots (b-a)$ each, where $a = \dfrac{b-1}{2m+2}$. Then, we choose $b-1$ to be a multiple of $2m+2$ (b will be odd). Hence, there will be types having from $b(m+1) - (b-1)/2$ to $(2m+1)b$ keys. Let $\beta = (m+1)(b-a)$. The transition matrix becomes

$m \setminus b$	3	6	9	12	15
1	.6071	.6154	.6184	.6200	.6210
	.8095	.8205	.8246	.8267	.8280
2	.6385	.6453	.6477	.6489	.6497
	.7662	.7743	.7772	.7787	.7796
3	.6532	.6587	.6606	.6616	.6622
	.7465	.7528	.7550	.7561	.7568
4	.6616	.6662	.6678	.6686	.6691
	.7352	.7402	.7420	.7429	.7434
5	.6672	.6711	.6724	.6731	.6735
	.7278	.7321	.7335	.7343	.7347

Table 6.2.3. Lower and upper bounds on $\overline{Un}_m$

$m \setminus b$	3	6	9	12	15	∞
1	.8862	.8570	.8465	.8411	.8378	.8244
2	.9229	.9031	.8963	.8928	.8906	.8820
3	.9416	.9267	.9216	.9190	.9175	.9111
4	.9530	.9411	.9370	.9350	.9337	.9287
5	.9607	.9507	.9474	.9456	.9446	.9405
10	.9784	.9729	.9711	.9702	.9696	.9674
20	.9886	.9858	.9848	.9843	.9841	.9829
30	.9923	.9903	.9897	.9894	.9892	.9883
40	.9942	.9927	.9922	.9920	.9918	.9912
50	.9953	.9941	.9937	.9935	.9934	.9929

Table 6.2.4. Lower bounds on $\overline{U\ell}_b$

$m \setminus b$	3	6	9	12	15	∞
1	.7873	.8055	.8118	.8149	.8168	.8244
2	.8253	.8502	.8599	.8650	.8682	.8820
3	.8442	.8726	.8840	.8902	.8941	.9111
4	.8554	.8861	.8986	.9054	.9097	.9287
5	.8629	.8951	.9084	.9156	.9202	.9405
10	.8799	.9157	.9306	.9389	.9441	.9674
20	.8897	.9275	.9435	.9523	.9579	.9829
30	.8931	.9317	.9480	.9570	.9628	.9883
40	.8948	.9338	.9503	.9595	.9653	.9912
50	.8959	.9351	.9518	.9610	.9668	.9929

Table 6.2.5. Lower bound on $\overline{Ut}_m$

$$H = \begin{bmatrix} -(\beta+1) & & & (2m+1)b+1 \\ \beta+1 & \ddots & & \\ & \ddots & \ddots & \\ & & (2m+1)b & -((2m+1)b+1) \end{bmatrix}$$

whose solution is:

$$p_i = \frac{\beta+1}{i+1}\, p_\beta\,,\ i = \beta+1, \cdots, (2m+1)b$$

with

$$p_\beta = \frac{1}{\beta+1}\,(H_{(2m+1)b+1} - H_\beta\,)^{-1}$$

Hence

$$Pr\{0\ splits\} = 1 - \frac{1}{b}\,(1 + \frac{1}{2m\ \ln 2}\,) + O(b^{-2}, m^{-2}\,)$$

$$Pr\{1\ split\} = \frac{\Psi(2m+1+1/b) - \Psi(m+1+1/b)}{b\ (H_{(2m+1)b+1} - H_\beta\,)} = \frac{1}{b} + O(b^{-2}, m^{-2}\,)$$

$$Pr\{2\ or\ more\ splits\} = \frac{1}{(2m+1)b\ (H_{(2m+1)b+1} - H_\beta\,)} = \frac{1}{2mb\ \ln 2} + O(b^{-2}, m^{-2}\,)$$

$$\bar{I}_{acc} \geq 2 + \frac{6m}{b} + \frac{1}{b}(3 + \frac{2}{\ln 2}) + O(1/mb)$$

$$\overline{n\ell}_m(N) = \frac{1}{(2m+1)b\ +1}\ \frac{(N+1)}{(H_{(2m+1)b+1} - H_\beta\,)} = (\frac{1}{2mb\ \ln 2} + O(b^{-2}, m^{-2}\,))\ (N+1)$$

$$(1 + \frac{1}{2m})\ \overline{n\ell}_m(N) \leq \bar{n}_m(N) \leq (1 + \frac{1}{m})\ \overline{n\ell}_m(N)$$

$$\overline{nk}_m(N) = \frac{\Psi(2m+2+1/b) - \Psi(m+1+1/b)}{b\ (H_{(2m+1)b+1} - H_\beta)}$$

$$\overline{Un}_m \geq \frac{(2m+1)b+1}{(2m+2)b}\ (\Psi(2m+2+1/b) - \Psi(m+1+1/b))$$

$$\overline{Un}_m \geq \ln 2 + \frac{1}{2m} + O(\frac{1}{mb})$$

$$\overline{U\ell}_b \geq \frac{H_{(2m+1)b+1} - H_\beta}{\Psi(2m+2+1/b) - \Psi(m+1+1/b)}$$

$$\lim_{b\to\infty} \overline{U\ell}_b \geq \frac{\ln(2m+1) - \ln(m+1/2)}{H_{2m+1} - H_m}$$

$$\lim_{b\to\infty} \overline{U\ell}_b = \lim_{b\to\infty} \overline{Ut}_m \geq 1 - \frac{1}{2m\ \ln 2} + O(m^{-2})$$

$$\lim_{m\to\infty} \overline{U\ell}_b = 1$$

$$\lim_{m\to\infty} \overline{Ut}_m \geq 1 - \frac{1}{b}\ (\ln^{-1}2 - 1) + O(b^{-2})$$

Here the upper bound in $\overline{U\ell}_b$ is $1 + \frac{1}{8(m+1)(2m+1)}$ times the lower bound. This is using the Kantorovich inequality and that $U_{\min} = (2m+1)/(2m+2)$. Tables 6.3.1 to 6.3.5 show some values of the measures.

7. Deepest Safe Nodes

This section is based in Eisenbarth *et al* [11]. An index node of a B$^+$-tree of order m is insertion safe if it contains fewer than $2m$ keys. Analogously, a bucket is insertion safe if it contains fewer than b keys. A safe node is the deepest one in a particular insertion path, if there are no safe nodes below it. When considering concurrency of operations on B$^+$-trees, one technique to permit simultaneous access to the tree by more than one process, is to lock the deepest safe node (*dsn*) on the insertion path.

m	b	3+2m	5+4m	7+6m	9+8m	11+10m
1	*Pr{0 splits}*	.8392	.9096	.9371	.9518	.9609
	Pr{1 split}	.0603	.0356	.0252	.0195	.0159
	Pr{2 or more splits}	.1006	.0548	.0377	.0287	.0232
	$\overline{I}_{acc}$	3.29	2.72	2.50	2.39	2.31
2	*Pr{0 splits}*	.9036	.9475	.9639	.9725	.9778
	Pr{1 split}	.0543	.0301	.0208	.0159	.0129
	Pr{2 or more splits}	.0421	.0224	.0153	.0116	.0094
	$\overline{E}[splits]$	.3069	.1645	.1126	.0855	.0693
	$\overline{I}_{acc}$	3.35	2.74	2.50	2.39	2.31
3	*Pr{0 splits}*	.9310	.9631	.9748	.9809	.9846
	Pr{1 split}	.0459	.0247	.0169	.0128	.0103
	Pr{2 or more splits}	.0232	.0122	.0083	.0063	.0050
	$\overline{E}[splits]$	.1329	.0705	.0480	.0364	.0291
	$\overline{I}_{acc}$	3.38	2.74	2.50	2.38	2.31
4	*Pr{0 splits}*	.9462	.9716	.9807	.9854	.9883
	Pr{1 split}	.0391	.0207	.0141	.0107	.0086
	Pr{2 or more splits}	.0147	.0077	.0052	.0039	.0031
	$\overline{E}[splits]$	.0848	.0447	.0303	.0228	.0182
	$\overline{I}_{acc}$	3.40	2.74	2.50	2.38	2.30
5	*Pr{0 splits}*	.9560	.9770	.9844	.9882	.9905
	Pr{1 split}	.0339	.0178	.0120	.0091	.0073
	Pr{2 or more splits}	.0101	.0053	.0036	.0027	.0022
	$\overline{E}[splits]$	.0623	.0327	.0221	.0167	.0135
	$\overline{I}_{acc}$	3.41	2.74	2.50	2.38	2.30

Table 6.3.1. Probabilities of 0 splits, 1 split, and 2 or more splits, upper bound on $E[splits]$ and lower bound on $\overline{I}_{acc}$.

$m \setminus b$	3+2m	5+4m	7+6m	9+8m	11+10m
1	.1414	.0793	.0551	.0422	.0342
	.1886	.1057	.0734	.0563	.0456
2	.0511	.0276	.0189	.0144	.0116
	.0613	.0331	.0227	.0173	.0139
3	.0266	.0141	.0096	.0073	.0059
	.0304	.0161	.0110	.0083	.0067
4	.0163	.0086	.0058	.0044	.0035
	.0182	.0095	.0065	.0049	.0039
5	.0111	.0058	.0039	.0029	.0024
	.0121	.0063	.0043	.0032	.0026

Table 6.3.2. Lower and upper bounds on $\dfrac{\overline{n}_m(N)}{(N+1)}$.

Let *Prob{dsn at* j^{th} *lowest level}* be the probability that the deepest safe node on a random search is located at the j^{th} $(j \geq 1)$ lowest level of a random B$^+$-tree with N keys. In the same way, let *Prob{dsn above* j^{th} *lowest level}* be the probability that the deepest safe node on a random search is located above the j^{th} $(j \geq 1)$ lowest level of a random B$^+$-tree with N keys. The 1^{st} lowest level is the buckets level.

It is clear that the probability that the deepest safe node is located at the j^{th} lowest level is equal to the probability that exactly $j-1$ splits occur on the $(N+1)^{st}$ random insertion. Therefore

$m \setminus b$	3+2m	5+4m	7+6m	9+8m	11+10m
1	.6136 .8182	.6184 .8246	.6204 .8272	.6214 .8286	.6221 .8295
2	.6463 .7755	.6492 .7791	.6503 .7804	.6509 .7811	.6513 .7815
3	.6606 .7550	.6624 .7571	.6631 .7579	.6635 .7583	.6637 .7585
4	.6684 .7427	.6697 .7441	.6701 .7446	.6703 .7448	.6705 .7450
5	.6733 .7345	.6742 .7355	.6745 .7358	.6747 .7360	.6748 .7361

Table 6.3.3. Lower and upper bounds on $\overline{Un}_m$.

$m \setminus b$	3+2m	5+4m	7+6m	9+8m	11+10m	∞
1	.8642	.8497	.8442	.8412	.8394	.8318
2	.9011	.8936	.8909	.8894	.8885	.8849
3	.9223	.9177	.9161	.9152	.9147	.9126
4	.9360	.9329	.9319	.9313	.9310	.9296
5	.9456	.9434	.9427	.9423	.9420	.9411
10	.9690	.9683	.9680	.9679	.9678	.9676

Table 6.3.4. Lower bounds on $\overline{U\ell}_b$.

$m \setminus b$	3+2m	5+4m	7+6m	9+8m	11+10m	∞
1	.8026	.8145	.8195	.8222	.8240	.8318
2	.8550	.8677	.8728	.8756	.8773	.8849
3	.8846	.8969	.9017	.9042	.9058	.9126
4	.9040	.9155	.9199	.9222	.9236	.9296
5	.9177	.9284	.9323	.9344	.9357	.9411
10	.9519	.9593	.9620	.9633	.9641	.9676

Table 6.3.5. Lower bounds on $\overline{Ut}_m$.

$$Prob\{dsn\ at\ j^{th}\ lowest\ level\} = Prob\{j{-}1\ splits\} \quad \text{and}$$
$$Prob\{dsn\ above\ j^{th}\ lowest\ level\} = Prob\{j\ or\ more\ splits\}.$$

For example, in a simple B^+-tree of order 2 and b=5, we have

$$Prob\{dsn\ at\ 1^{st}\ lowest\ level\} = 0.730\ ,$$
$$Prob\{dsn\ at\ 2^{nd}\ lowest\ level\} = 0.198 \text{ and}$$
$$Prob\{dsn\ above\ 2^{nd}\ lowest\ level\} = 0.108\ .$$

In other words, 89% of the time we lock a bucket or an index node in the last level. For a B^+-tree of greater order and bucket size the solution analyzed here will lock a bucket most of the time. This shows that complicated solutions for concurrency are rarely of benefit.

8. Conclusions

The most important measure for comparing the performance of B^+-trees is memory utilization, mainly represented by bucket memory utilization. Other measures, such as the number of comparisons and the number of accesses to secondary memory, turn out to be similar for all cases ($O(\log_m(N))$). Figure 8.1 consists of a graph of $\overline{U\ell}_b$ for m=1, for all the techniques analyzed. The main conclusion that can be drawn from the graph is that the simplest overflow technique (B*-trees), gives a memory utilization of over 81%.

For the other techniques, the improvement is much less. Furthermore, the cost of an insertion increases by adding an $O(m)$ term to the number of accesses to secondary memory. Consequently, these more sophisticated techniques may not be warranted since, even for a small value of m, the number of accesses may double. For example, in a tree with m=2 and b=6 the expected insertion cost is 2.46. This cost is near 3 for B*-trees and over 4 for the other overflow techniques. In other words, the insertion time is doubled, resulting in only a 25% increase in the storage utilization.

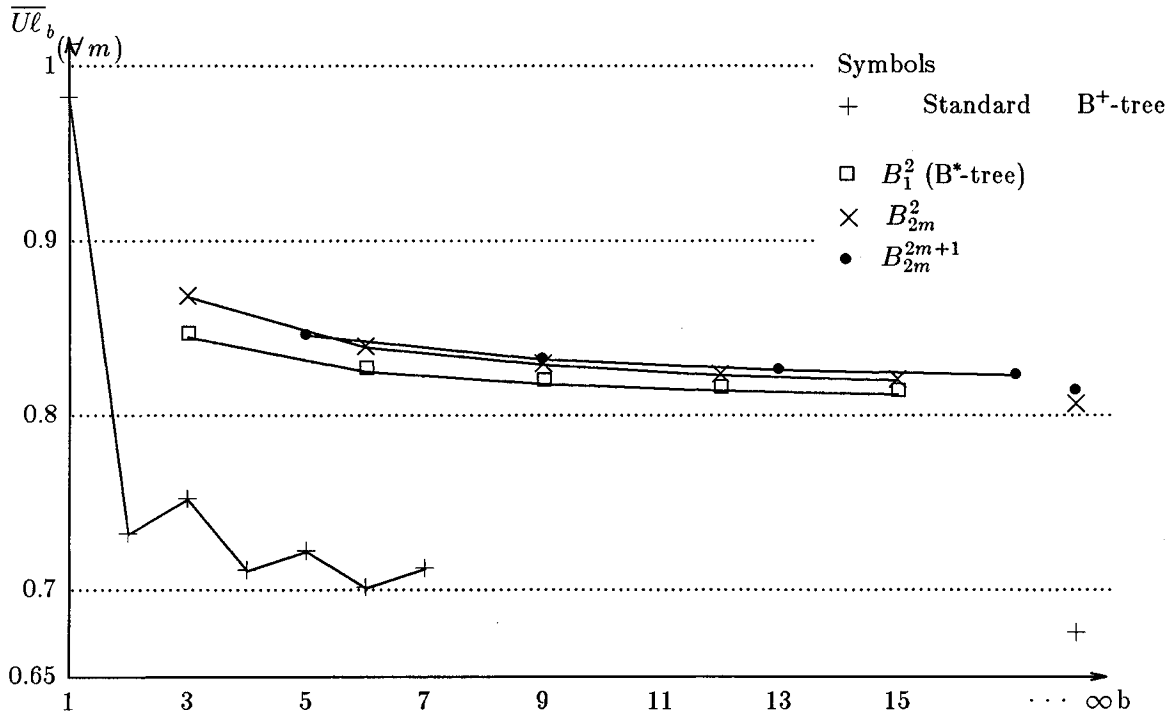


Figure 8.1. Lower Bound on the expected bucket memory utilization (m=1).

The memory utilization in the index does not improve by using a more elaborate overflow technique at the bucket level and remains constant at ln2 (as in B-trees). See figure 8.2 (m = 1). This value could be improved if overflow techniques were also used in the index. Figure 8.3 shows the graph of total memory utilization for m = 1.

In comparison with standard B-trees, the performance of B^+-trees is similar for asymptotic results as $b \to \infty$ [11]. For small values of b, however, memory utilization is better. Furthermore, when overflow techniques are used, as can be seen, for example, by comparing with the case m=1 [3]. Memory utilization for the B* case seems to be a little better than 81% [17], for large m. Comparing with dense multiway trees, a B^+-tree proves to be better; for example for m=2 a B*-tree with b=3 has an expected utilization of 87% in the buckets whereas for a dense multiway tree the value is 82% [10].

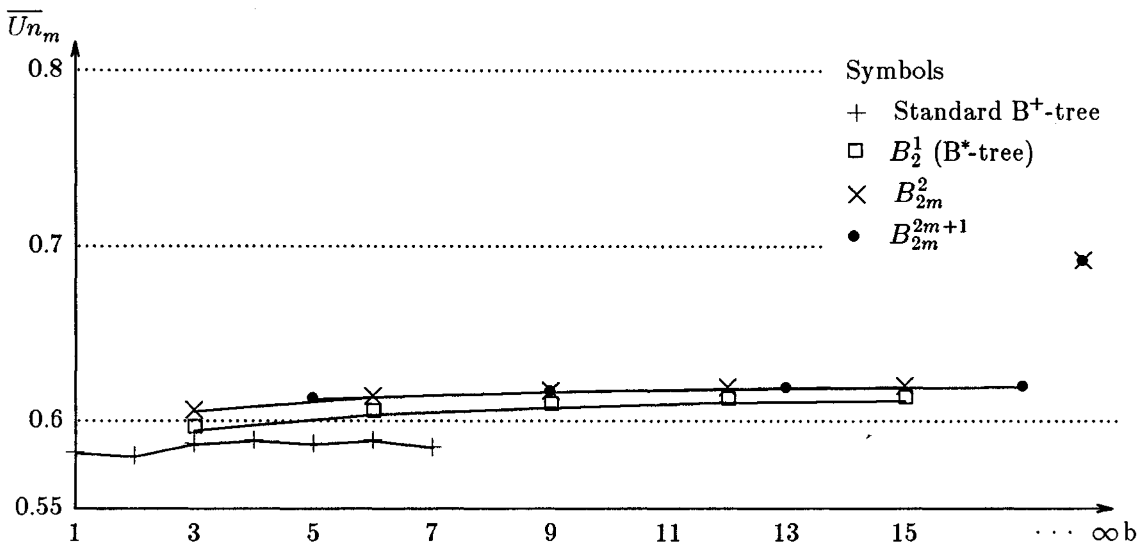


Figure 8.2. Lower bound on the expected memory utilization in the index nodes (m=1).

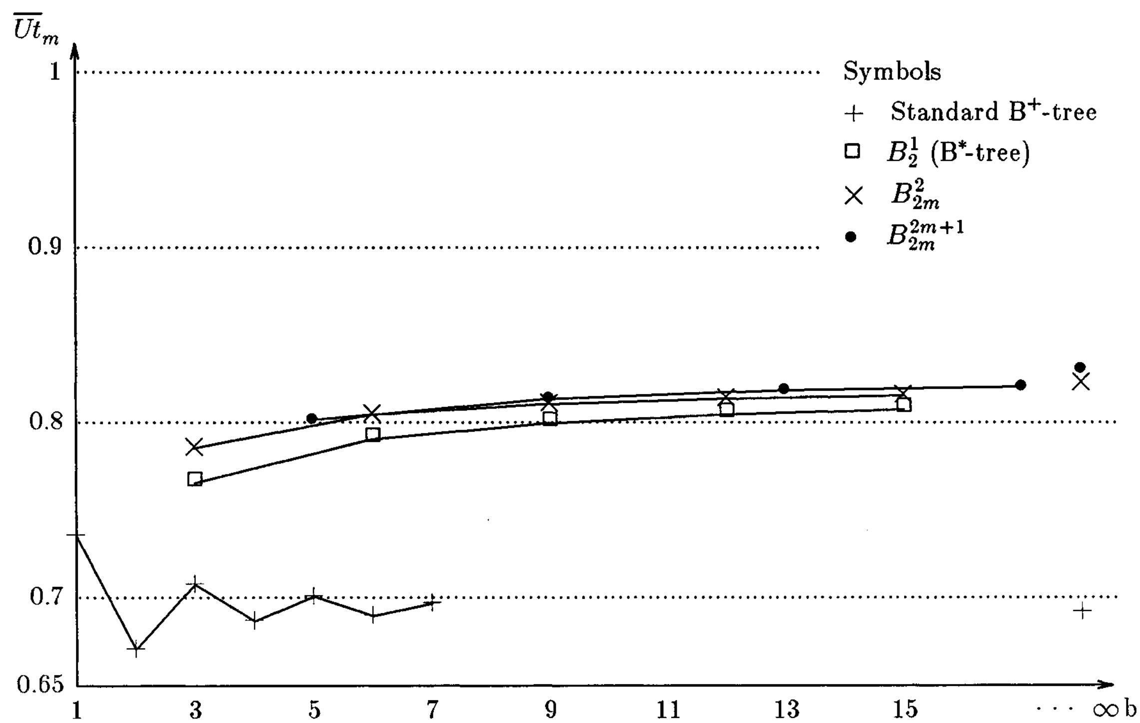


Figure 8.3. Lower bound on the expected total memory utilization (m=1).

Our analysis, albeit different, shows similar results to those of Kuspert [16] as far as memory utilization is concerned. Our analysis, however, yields other measures that go unmentioned in that paper. Table 8.1. shows the values obtained by Kuspert contrasted with ours.

Split factor	Kuspert [16]	This work		
		Lower bound	Upper bound	m,ℓ
1	.6931	.6931	.7798	$\geq$1,0
2	.8109	$\geq$.8170	$\geq$.8510	$\geq$2,2
		$\geq$.8244	$\geq$.8588	$\geq 1,2m$
3	.8630	.8318	.8491	1,2
5	.9116	.8849	.8923	2,4
7	.9347	.9126	.9167	3,6
9	.9482	.9296	.9322	4,8
11	.9571	.9411	.9429	5,10
13	.9634	.9493	.9506	6,12

Table 8.1. Memory utilization versus split factor.

The main difference between Kuspert's study and ours, is that the former obtains expected values of the utilization at the bucket level (assuming that $1/\mathrm{E}(x) = \mathrm{E}(1/x)$), and we obtain lower and upper bounds. Also, his analysis does not consider the order of the tree. Kuspert's values are higher for split factors greater than two, because are for large order m.

A B-tree analysis including deletions, based on a continuous model was developed by Quitzow and Klopprogge [20]. They made the simplifying assumption that deletions preserve the randomness of the tree, which is clearly not entirely correct. If only insertions occur, their analysis yields figures similar to ours (69.2%). When an overflow technique is used, which instead of splitting, first tries to compensate the contents of the bucket with their brothers, the value becomes 91.8% . If the insertion rate equals that of deletions, these figures decrease to 59.2% and 82.6% respectively. Hence, deletions appear to worsen the behaviour of B-trees.

The theoretical values obtained from the analysis are consistent with empirically observed results. Figures between 67% and 71% have been obtained for the simple case [2]; around 85% for the B* case [5]; and between 81% and 82% for variable-length keys [18].

Finally, it is important to note, that all the results are valid for a B-tree of order m when $b=2m+1$ in a B$^+$-tree (when the size of a bucket is equal to the size of an index node). Then, the second order analysis for simple B$^+$-trees, and for the B* and B_{2m}^{2m+1} overflow techniques, are also new results for B-trees.

The effect of the initial state is studied in [19]. The iteration of equation (3.2) shows similar results for B$^+$-trees, i.e., oscillations before reaching the steady state solution. The oscillations decrease when b or m is increased and depends on the initial state.

We have shown how fringe analysis can be applied to B$^+$-trees to study the effects of the order of the tree and the bucket size on several performance measures. The results are consistent with those of previous theoretical and empirical analysis. Problems that remain open are the influence of variable-length keys and the expected value of the height of the tree, and, finally, the problem of the effect of deletions on performance for search trees in general.

Acknowledgments

The author wishes to acknowledge the helpful comments of Gaston Gonnet and Per-Ake Larson.

References

1. Abramowitz, M. and Stegun, I.: *Handbook of Mathematical Functions*. New York: Dover, 1972.
2. Arnow, D. and Tenenbaum, A.: An Empirical Comparison of B-Trees, Compact B-Trees and Multiway Trees, *ACM SIGMOD* **14**, 2 (1984), 33-46.
3. Baeza-Yates, R. and Poblete, P.: Reduction of the Transition Matrix of a Fringe Analysis and Its Application to the Analysis of 2-3 Trees (in Spanish), 5^{th} International Conference in Computer Science, Santiago, Chile (1985), 56-82.
4. Bayer, R.: Binary B-Trees for Virtual Memory, Proceedings of the 1971 ACM SIGFIDET Workshop, San Diego (1971), 219-235.
5. Bayer, R. and McCreight, E.: Organization and Maintenance of Large Ordered Indexes, *Acta Informatica* **1**, 4 (1972), 173-189.
6. Bayer, R. and Unterauer, K.: Prefix B-Trees, *ACM Transactions on Database Systems* **2**, 1 (1977), 11-26.
7. Brown, M.: Some Observations on Random 2-3 Trees, *Information Processing Letters* **9**, 2 (1979), 57-59.
8. Clausing, A.: Kantorovich-Type Inequalities, *The American Mathematical Monthly* **89**, 5 (1982), 314-330.
9. Comer, D.: The Ubiquitous B-Tree, *Computing Surveys* **11**, 2 (1979), 121-137.
10. Culik, K., Ottmann, T. and Wood, D.: Dense Multiway Trees, *ACM Transactions on Database Systems* **6**, 3 (1982), 486-512.
11. Eisenbarth, B., Ziviani, N., Gonnet G.H., Mehlhorn K. and Wood, D.: The Theory of Fringe Analysis and Its Application to 2-3 Trees and B-Trees, *Information and Control* **55**, 1-3 (1982), 125-174.
12. Gupta, G. and Srinivasan, B.: Approximate Storage Utilization of B-Trees, *Information Processing Letters* **22**, 4 (1986), 243-246.
13. Jensen, L.W.V.: Sur les Fonctions Convexes et les Inegalites Entre les Valeurs Moyannes, *Acta Mathematica* **30** (1906), 175-193.
14. Knuth, D.E.: *The Art of Computer Programming, Vol. 1: Fundamental Algorithms,* 2nd ed. Reading, Mass.: Addison-Wesley, 1973.
15. Knuth, D.E.: *The Art of Computer Programming, Vol. 3: Sorting and Searching,* 1st ed. Reading, Mass.: Addison-Wesley, 1973.
16. Kuspert, K.: Storage Utilization in B*-Trees with a Generalized Overflow Technique, *Acta Informatica* **19**, 1 (1983), 35-56.
17. Leung, C.: Approximate storage utilisation of B-Trees: A simple derivation and generalizations, *Information Processing Letters* **19** (1984), 199-201.
18. Mc Creight, E.: Pagination of B*-Trees with Variable-Length Records, *Communications of the ACM* **20**, 9 (1977), 670-674.
19. Nakamura, T. and Mizoguchi, T.: An Analysis of Storage Utilization Factor in Block Split Data Structuring Scheme; Proceedings VLDB, Berlin, 4 (Sept 1978), 489-495.
20. Quitzow, K. and Klopprogge, M.: Space Utilization And Access Path Length in B-Trees, *Information Systems* **5**, 1 (1980), 7-16.
21. Wright, W.: Some Average Performance Measures for the B-Tree, *Acta Informatica* **21**, 1 (1985), 541-557.
22. Yao, A.: On Random 2-3 Trees, *Acta Informatica* **9**, 159-170 (1978).
23. Ziviani, N., Olivié, H. and Gonnet, G.: The Analysis of an Improved Symmetric Binary B-Tree Algorithm, *The Computer Journal* **28**, 4 (1985), 417-425.