INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)”. If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of “sectioning” the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106
A PROBABILISTIC ANALYSIS OF A CLASS
OF RANDOM TREES

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

by

Hosam M. Mahmoud

* * * * *

The Ohio State University
1983

Reading Committee:

Dr. Bruce W. Weide
Dr. Venkataraman Ashok
Dr. Boris Pittel

Approved by

Dr. Bruce W. Weide
Adviser
Department of Computer
and Information Science
To my Parents
ACKNOWLEDGEMENTS

I would like to express my gratitude to all the members of my reading committee. Dr. Ashok gave me all kinds of support: academic, moral, and financial. Dr. Weide helped me define the problem and explained to me many of its intriguing aspects. We spent countless hours together trying to crack the problem. At moments of despair, when I was left alone with a ferocious problem, Dr. Pittel guided me to the way out and helped me so much with the asymptotic analysis, an area about which I had known so little before starting this work.
VITA

April 16, 1954 ................................................. Born in Cairo, Egypt

July, 1976 ...................................................... B. Sc.,
Department of Electrical
Engineering,
Cairo University
Cairo, Egypt

1977.............................................................. Mandatory military service,
Teaching Associate,
Military Technical College
Cairo, Egypt

July, 1979 ...................................................... B. Sc.,
Department of Mathematics,
Cairo University
Cairo, Egypt

September, 1979 - August, 1983 ...................... Teaching Associate,
Computer Science Department,
The Ohio State University,
Columbus, Ohio

Autumn, 1980 ................................................ M. S.,
Computer Science Department,
The Ohio State University,
Columbus, Ohio

Summer, 1983 ............................................... Research Associate,
Computer Science Department,
The Ohio State University,
Columbus, Ohio

PUBLICATIONS

"On the Most Probable Shape of a Search Tree Grown from a Random

iv
FIELDS OF STUDY

Major Field: Computer Science

Studies in Algorithms and Data Structures, Prof. Bruce Weide
Studies in Automata Theory, Prof. Jerome Rothstein
Studies in Combinatorics, Prof. D. RayChaudhuri
TABLE OF CONTENTS

Page

DEDICATION ................................................. 11

ACKNOWLEDGEMENTS ..................................... 111

VITA ................................................................. iv

LIST OF TABLES .............................................. vii

LIST OF FIGURES ............................................. ix

CHAPTER

I. INTRODUCTION ............................................. 1
   1.1 Binary search trees defined................................. 2
   1.2 The path lengths in a tree ................................. 5
   1.3 What is already known about the height of trees... 6
   1.4 Outline of the thesis ...................................... 7

II. SOME PROPERTIES OF THE BINARY SEARCH TREE ....... 9
   2.1 Correspondence between permutations and trees... 9
   2.2 The expected number of leaves in a BST ............ 11
   2.3 The expected shape of a BST .......................... 14
   2.4 The expected level of the leftmost external node 18
   2.5 The expected internal and external path lengths... 20

III. THE SHORTEST AND LONGEST PATHS IN A BST:
     A COMBINATORIAL APPROACH ............................ 25
     3.1 Permutations and the height of a BST .............. 26
     3.2 Perfect trees ........................................... 29
     3.3 Permutations and trees of certain path lengths.. 32
     3.4 The exponential generating function of the
         sequences T(n,k) and t(n,k) ......................... 34

IV. THE SHORTEST AND LONGEST PATHS IN A BST:
    AN ASYMPTOTIC APPROACH ............................... 37
    4.1 Preliminaries and results .............................. 40
    4.2 Auxiliary combinatorial relations .................. 42
    4.3 Asymptotic estimates of $E_{nk}$ .................... 45
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Proof of theorems 4.1, 4.2</td>
</tr>
<tr>
<td>4.5</td>
<td>A remark on the distribution of endpoints</td>
</tr>
<tr>
<td>V.</td>
<td>m-WAY TREES AND THEIR BALANCING</td>
</tr>
<tr>
<td>5.1</td>
<td>m-way search trees defined: an insertion model</td>
</tr>
<tr>
<td>5.2</td>
<td>Off-line balancing</td>
</tr>
<tr>
<td>5.3</td>
<td>On-line balancing</td>
</tr>
<tr>
<td>VI.</td>
<td>ASYMPTOTIC ANALYSIS OF m-WAY TREES</td>
</tr>
<tr>
<td>6.1</td>
<td>Results and interpretations</td>
</tr>
<tr>
<td>6.2</td>
<td>Proofs</td>
</tr>
<tr>
<td>VII.</td>
<td>CONCLUSION AND TOPICS FOR FURTHER RESEARCH</td>
</tr>
</tbody>
</table>

APPENDIXES

A. COMPUTER RESULTS CONCERNING THE HEIGHT OF A BST | 96 |
B. COMPUTER RESULTS CONCERNING THE SHORTEST PATH FROM THE ROOT TO A LEAF NODE IN A BST | 101 |

BIBLIOGRAPHY | 106 |
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. THE AVERAGE HEIGHT FOR SOME CLASSES OF TREES.</td>
<td>6</td>
</tr>
<tr>
<td>2. THE ROOTS OF (6.5) FOR SOME SMALL VALUES OF m</td>
<td>72</td>
</tr>
<tr>
<td>3. THE AVERAGE AND VARIANCE OF THE HEIGHT OF A BST ON n NODES</td>
<td>96</td>
</tr>
<tr>
<td>AS COMPARED TO ln n</td>
<td></td>
</tr>
<tr>
<td>4. THE AVERAGE AND VARIANCE OF THE SHORTEST PATH FROM THE ROOT</td>
<td>101</td>
</tr>
<tr>
<td>TO A LEAF NODE IN A BST ON n NODES AS COMPARED TO ln n.</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>A permutation and its tree.</td>
</tr>
<tr>
<td>2</td>
<td>The extended tree corresponding to the unextended tree of figure 1.</td>
</tr>
<tr>
<td>3</td>
<td>Finding a permutation for a given tree.</td>
</tr>
<tr>
<td>4</td>
<td>The trees of the permutations on (1,2).</td>
</tr>
<tr>
<td>5</td>
<td>Two types of non-terminal nodes.</td>
</tr>
</tbody>
</table>
| 6      | Two cases: (a) one in the root  
(b) one in the left subtree            | 19   |
<p>| 7      | i goes into the root node                                                  | 22   |
| 8      | Trees with extremal path lengths.                                          | 25   |
| 9      | Two permutations that give the same BST                                    | 26   |
| 10     | The extended BST corresponding to the above permutations.                 | 26   |
| 11     | A legal arrangement of XUY.                                                | 28   |
| 12     | A perfect tree on (n) nodes ((n = 2^m - 1)).                         | 30   |
| 13     | i is in the root and the left and right subtrees are constructed from the sets L and R | 32   |
| 14     | The graph of the function (g(a)).                                       | 48   |
| 15     | The sequence of ternary trees constructed from the permutation ((4,2,3,5,7,6,1)). | 54   |
| 16     | m-way trees with extremal path lengths.                                    | 55   |
| 17     | An off-line balancing algorithm                                            | 57   |</p>
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Local rotation at the fringe nodes to make the median key the root of the fringe subtree</td>
<td>60</td>
</tr>
<tr>
<td>19</td>
<td>The tree constructed from w (a) using direct insertion (b) using the fringe heuristic</td>
<td>62</td>
</tr>
<tr>
<td>20</td>
<td>COUNTER changes according to the number of subtrees of each node</td>
<td>63</td>
</tr>
<tr>
<td>21</td>
<td>Direct insertion and the fringe heuristic in the worst case</td>
<td>64</td>
</tr>
<tr>
<td>22</td>
<td>An m-ary tree with subtrees of sizes $i_1, i_2, \ldots, i_m$</td>
<td>75</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

In his milestone book, The Art of Computer Programming, Knuth [1] begins his chapter on trees with "We now turn to a study of trees, the most important nonlinear structure arising in computer algorithms." Indeed, he is right, as trees are accepted as a universal representation of data sets of all types. Actually, trees arise in all aspects of computer science, not only in algorithms. Compilers build parse trees of the programs compiled [2], [3]. Trees are used to maintain files and analyze programs [1], [4], [5].

Trees represent pictures in image processing [6], [7]. They are also used in VLSI computer-aided design to represent IC layout [8]. Trees are used to represent algebraic expressions [9]. In computer architecture, trees present new trends; the tree machine is one example [10], [11].

It would take a complete volume to present and explain all of the different uses of trees. Our primary intent in this thesis is to study a very important class of trees called search trees. We start with the definition of an important subclass of this family called the binary search trees.
1.1 Binary search trees defined

In many practical situations, we construct trees from input sequences on line (as they progress in time). It is a fairly realistic assumption that all input sequences are equally likely.

Consider a permutation $w(n) = (w(1), w(2), \ldots, w(n))$ of the integers $1, 2, \ldots, n$. The computer reads $w(1), w(2), \ldots, w(n)$ each in turn and constructs a binary search tree (BST from now on) $t_n = t(w(n))$, which is defined recursively, essentially in the same way as in [4]. Namely, a BST $t_n$ is either empty or each node is labeled with an integer $i, 1 \leq i \leq n$, subject to the following:

1. all the labels in the left subtree are less than the root label;
2. all the labels in the right subtree are greater than the root label; and
3. the left and the right subtrees are also binary search trees.

In what follows we informally illustrate an algorithm to construct $t_n$ from $w(n)$ with an example. Consider the permutation:

$$w(n) = (4, 2, 3, 6, 5, 1).$$

Four appears first, so it constitutes the root, and the tree becomes $\circ$. Two comes next, and since $2 < 4$, it goes to the left. The tree becomes $\circ$.

Three is next. Since $3 < 4$, it goes to the left subtree where it is compared with 2. Thus it goes to the right subtree rooted at 2, and we
get the tree

The construction proceeds in this manner until we end up with the full tree shown in figure 1.

We shall refer to this algorithm as the BST-algorithm. It is interesting to notice that more than one permutation may give the same tree under this construction scheme. For example, the permutation

$$w(6) = (4, 6, 2, 1, 3, 5)$$

will give the same tree as in Figure 1.1.

We may relax the constraint that $$w^{(n)}$$ is an n-tuple of integers since the algorithm is well defined for any n-set with a total order defined on it (a chain). Furthermore, two n-long sequences $$w_1^{(n)}$$ and
\(w_2^{(n)}\) give rise to the same step-by-step realization of the algorithm provided that their integer valued vectors of sequential ranks are identical. We say

\[ r^{(n)} = (r_1, r_2, \ldots, r_n) \]

is the vector of sequential ranks of a sequence \(w^{(n)}\) if \(w^{(n)}(k)\) is the \(r_k\)th in the chain formed by the elements \(w(1), w(2), \ldots, w(k), 1 \leq r_k \leq k\).

The vector

\[ r^{(n)} = (r_1, r_2, \ldots, r_n) \]

becomes the vector of absolute ranks if \(w^{(n)}(k)\) is the \(r_k\)th in the chain formed by the elements \(w(1), w(2), \ldots, w(n), 1 \leq r_k \leq n\). For example, the sequence of words 'END', 'IF', 'DO' defines the vector of sequential ranks \((1, 1, 2)\), and the vector of absolute ranks \((2, 1, 3)\) when the ordering relation is the usual ASCII collating sequence. The same vector of sequential ranks is obtained from the permutation \((2, 3, 1)\).

It is customary to work with \(T_n\), the extended BST which is obtained from a given BST \(t_n\) by adding the exact number of external nodes (square) as "children" of internal nodes (circular) with degrees less than two to make all internal nodes of outdegree two. The extended BST of the tree of figure 1 is shown below.

Figure 2: The extended tree corresponding to the unextended tree of figure 1.
For the rest of this work, when we say a random BST, we mean the randomness induced on the space of trees by considering all permutations as equally likely inputs to the BST-algorithm. When we say random binary trees, we mean the randomness induced by considering all binary trees as equally likely.

1.2 The path lengths in a tree

Two characteristics of trees are fundamental in determining the best and worst case time and space requirements for many of the tree applications. These two characteristics are the shortest and the longest paths of a tree. The longest path from the root to a leaf node (also known as the height or depth) is very important for all recursive algorithms on a tree, as it controls the memory required for the stack when the recursion gets to the longest branch of the tree. The depth of the stack of recursion is the depth of the tree for this worst path. Examples of such recursive algorithms are as follows:

1. Traversing trees in Preorder, Inorder, or Postorder [4].
2. Quick sort [1].

Also, the searching time for any key in the tree is $O(h)$, if $h$ is the height of the tree. The shortest path is the other end of the spectrum, and it represents the best case performance of some tree algorithms. For example, the time needed to insert or search for a record in a tree is proportional to the length of its shortest path in the best case.

The longest path from the root to a leaf node was studied for several classes of trees. The shortest path does not seem to have had
much attention.

1.3 What is already known about the height of trees

As we mentioned in the previous section, the height of trees was the focus of attention in studying random trees. Finding the average height of a class of random trees seems to be a hard problem, and only asymptotics of the height are known for most classes. The heights of the following classes were studied:

- binary trees [12]
- planted planar trees [13]
- labeled trees [14].

Table 1 summarizes the results known about the height of trees under different probability measures.

**Table 1**

<table>
<thead>
<tr>
<th>CLASS</th>
<th>AVERAGE HEIGHT</th>
<th>DESCRIPTION OF THE CLASS</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended binary</td>
<td>$\sim 2\sqrt{n}$</td>
<td>n internal nodes and $n + 1$ external nodes</td>
<td>Flajolet and Odyzko [12]</td>
</tr>
<tr>
<td>Planar</td>
<td>$\sim \sqrt{n}$</td>
<td>arbitrary node degrees</td>
<td>De Bruijn, Knuth and Rice [13]</td>
</tr>
<tr>
<td>Labeled</td>
<td>$\sim 2\sqrt{n}$</td>
<td>all $n^{n-2}$ trees which are sub-graphs of $K_n$ (the complete graph on $n$ nodes) are equally likely. Many of these trees are isomorphic.</td>
<td>Renyi and Szekeres [14]</td>
</tr>
</tbody>
</table>
All these families were shown to have $O(\sqrt{n})$ height. The height of BST, when all input sequences of length $n$ are equally likely, was shown by Robson [15] to be $O(\log n)$, with high probability. This result tells us that this probability model has a different nature.

1.4 Outline of the thesis

In this work, we study the class of search trees constructed from finite and infinite random input sequences when all such sequences are equally likely.

In Chapter II, we explain the basic techniques, and they include recurrence relations, probabilistic estimates, and generating functions. We show the effectiveness of these methods combined together in deriving some known results for binary trees. We give proofs different from the original using these techniques. We also use these methods to obtain new results concerning the relationship between trees and permutations, and we find the expected number of endpoints (leaves) in the unextended BST and use it to draw a conclusion about the expected distribution of vertices in BSTs.

In Chapter III, we use a combinatorial approach to find the recurrence relations governing the number of permutations giving trees of certain shortest and longest paths from the root to a leaf node. We use these recurrences to establish the average values of the height and the length of the shortest path from the root to a leaf node. A lemma established in this chapter enables us to find closed form solutions for a special subclass of BSTs, the class of perfect trees.
Numerical experiments (Appendices A and B) on the recurrences of Chapter III furnish some evidence that all paths from the root to a leaf node, in a BST with \( n \) nodes, have lengths \( O(\log n) \), and in Chapter IV, we perform an asymptotic analysis on the BSTs based on Robson's result [15] and the numerical experiments. We derive results of the type in probability and almost surely about paths from the root to leaves in a random BST.

In Chapter V, we extend the idea to \( m \)-ary trees constructed from random permutation. We start with the formal definition of this class of random trees, then we provide two balancing schemes. One is an off-line algorithm which always produces a complete tree (in a complete tree, the heights of any two subtrees of any node differ by one at most). The other scheme is an attempt to adapt the fringe heuristic [16] to process \( m \)-ary trees.

Chapter VI contains generalizations of the result of Chapter IV to \( m \)-ary trees. Furthermore, asymptotic estimates are developed for the number of comparisons needed to insert keys in \( m \)-ary search trees, an issue which does not arise in the binary case (\( m = 2 \)).

Chapter VII includes a summary and topics for further research.
In this chapter, we shall study some properties of the class of binary search trees constructed from a random sequence of \( n \) elements using the BST-algorithm when all \( n! \) sequences are equally likely (\( n \) will always be the number of nodes in the tree). Studying an additive property, i.e., a property in which every node has a contribution to be added to get the desired property, we shall always start with the expected value of a random variable directly or indirectly related to it. This approach is advantageous over direct combinatorial methods for additive properties due to the linearity of \( E \), the expectation operator, since we can simply add contributions of the right and left subtrees as well as the contribution of the root.

2.1 Correspondence between permutations and trees

The algorithm BST of section 1.1 constructs a tree for any given permutation. The converse is also an interesting question: given a tree, is there a permutation that will produce it if the BST-algorithm is applied? In other words, is every tree of the \( \frac{1}{n+1} \binom{2n}{n} \) trees on \( n \) nodes "realizable" by some permutation of \( \{1, 2, \ldots, n\} \)? Or do permutations "skip" some of these trees? The following simple lemma
answers this question.

Lemma 2.1

Every tree is realizable by at least one permutation.

Proof

We prove this lemma by induction on n. If n = 1, the tree is simply a single node, label it 1 to get \(1\); the permutation is \(1\). Assume that for every tree on \(n\) nodes, \(n < k\), there is a permutation on \((1, 2, \ldots, n)\) realizing it. Now let \(t_k\) be a tree with \(k\) nodes, and let \(t_{k-1}, t_{k-1}^*\) be its left and right subtrees. By the induction hypothesis, there exist permutations \(\pi_1, \pi_2\) on \((1, 2, \ldots, k-1)\) realizing the two subtrees. A permutation that will produce this tree on using the BST-algorithm is \(\pi = (1, \pi_1, \pi_2')\), where \(\pi_2'\) is \(\pi_2\) with all its elements increased by 1. In \(\pi\), the first element is 1, thus it will label the root, all the elements of \(\pi_1\) are in \((1, 2, \ldots, k-1)\), i.e., less than 1, therefore will go to the left and will produce \(t_{k-1}\), and all the elements of \(\pi_2'\) are greater than 1, so they will be guided to the left. But \(\pi_2'\) and \(\pi_2\) have the same vector of sequential ranks, so \(\pi_2\) will produce a subtree identical to that produced by \(\pi_2\), namely the tree \(t_{k-1}\).

Q.E.D.

The constructive proof of the above lemma can be used as the basis for a recursive algorithm to find a permutation for any given tree. Actually the proof constructs the permutation which will result from
traversing the tree using Preorder [4].

Example

Find a permutation that produces the tree of figure 3.

For the right subtree, \( n = 3, i = 2 \), and it has the subtrees \( t_1 \) and \( t_1 \); these are produced by \( \pi_1 = (1), \pi_2 = (1) \), so \( \pi_2' = (3) \), and the permutation \( \pi = (2, 1, 3) \) produces the right subtree. Similarly, the permutation \( \psi = (2, 1) \) produces the left subtree. For the whole tree, \( n = 6, i = 3 \). Let \( \pi' \) be \( \pi \) with all its elements increased by 3. Now the permutation \( (3, \psi, \pi') = (3, 2, 1, 5, 4, 6) \) produces the tree.

In fact, the constructive proof of the above lemma can be used to find all the permutations corresponding to a given tree by interleaving \( \pi_1 \) and \( \pi_2' \) in all possible ways. Lemma 3.1 in the next chapter explains how this can be done.

2:2 The expected number of leaves in a BST

The lemma established in this section will help us understand the structure of a BST.
Lemma 2.2

Let $e_n$ be the random number of endpoints (leaves) in an unextended random BST, and let $E(e_n)$ be the expected value of this random variable. We denote $E(e_n)$ as $E_n$ for brevity. Then

$$E_n = \frac{(n+1)}{3}, \text{ } n > 2, \text{ and } E_2 = 1, E_1 = 1.$$

Proof

When $n = 1$, the tree is a single node which is also an endpoint, so $E_1 = 1$. When $n = 2$, we have two permutations giving the trees of figure 4.

![Figure 4](image)

Therefore, $E_2 = \sum_{i=0}^{\infty} i \cdot \text{probability of } i \text{ endpoints}$

$= 0 + 1 \cdot (2/2) + 0 + 0 + \ldots$

$= 1.$

For $n > 2$, the conditional expected value when $i$ appears in the root of the tree ($i$ is the first in the permutation) is given by the following:

$$E(e_n | i \text{ is in the root}) = E(e_{i-1} + e_{n-i} | i \text{ is in the root}).$$
since the number of endpoints in the tree is the sum of the number of endpoints in the left and right subtrees. To get the unconditional expectation, multiply the expected value for each $i$ by the probability that $i$ will appear in the root, then sum over $i$,

$$E(e_n) = \sum_{i=1}^{n} P(i \text{ is in the root}) \cdot E(e_{i-1} + e_{n-i} | i \text{ is in the root}).$$

Our basic assumption that all $n$-long sequences are equally likely implies that the probability that any integer will be first in the permutation (i.e., will go into the root node) is $\frac{1}{n}$. Using linearity of the $E$ operator and that the left and right subtrees have the same distribution as the tree itself, we arrive at the following:

$$(2.1) \quad E_n = \frac{1}{n} \sum_{i=1}^{n} (E_{i-1} + E_{n-i}).$$

The second term of the summation is symmetric with the first (it lists the same terms backwards) and can be made identical to the first by replacing $i$ with $n-i+1$ to get

$$(2.2) \quad nE_n = 2 \sum_{i=1}^{n} E_{i-1}, \quad n > 2.$$  

To solve this recurrence, first rewrite (2.2) with $n+1$ instead of $n$ to get

$$(2.3) \quad (n+1)E_{n+1} = 2 \sum_{i=1}^{n+1} E_{i-1}.$$  

Now subtract (2.2) from (2.3) to get

$$E_{n+1} = \frac{n+2}{n+1} E_n,$$
or

\[(2.4) \quad E_n = \frac{(n+1)}{n} E_{n-1}, \quad n > 2.\]

The recurrence is now in a form suitable for iterative substitution.

\[
E_n = \frac{n+1}{n} E_{n-1} \\
= \frac{n+1}{n} \frac{n}{n-1} E_{n-2} \\
\vdots \\
= \frac{n+1}{n} \frac{n}{n-1} \frac{n-1}{n-2} \cdots \frac{5}{4} \frac{4}{3} E_2 \\
= \frac{n+1}{3}
\]

Q.E.D.

The methods used to reach (2.4) from (2.1) will be used time and again throughout this work, and we shall not repeat them, but we shall rather refer to the proof of the above lemma whenever needed. All of these techniques are well known and well used in connection with trees (see [1], [4]).

2.3 The expected shape of a BST

In this section, we investigate the shape of a binary tree. By "shape" we mean the distribution of vertices in the tree. Simple well-known results from graph theory are used as we consider the undirected graph underlying our directed tree. The leaves are nodes of degree one, and non-terminal nodes may have degrees two or three as in figure 5. The root may have degree one, however, since it has no "parent" in the
Let $V_i$ be the set of vertices of degree $i$, $i = 1, 2, 3$, and let $v_i$ be the expected number of vertices of degree $i$, $i = 1, 2, 3$, (i.e., $v_i = E(|V_i|)$) in a BST on $n$ nodes. Then

$$v_1 \sim v_2 \sim v_3 \sim \frac{n}{3}.$$ 

**Proof**

For every permutation, we have

$$|V_1| = \begin{cases} 
  e_n + 1 & \text{if the root has degree 1}, \\
  e_n & \text{if the root has degree 2}.
\end{cases}$$

That is, for every permutation

$$e_n \leq |V_1| \leq e_n + 1.$$
Averaging over all permutations:

\[ E(e_n) \leq E(|V_1|) \leq E(e_n + 1) , \]
\[ E_n \leq v_1 \leq E_n + 1 , \]

where \( e_n , E_n \) are as in the previous lemma. Hence,

\[ (2.5) \quad \frac{(n+1)}{3} \leq v_1 \leq \frac{(n+1)}{3} + 1 , \]

or

\[ v_1 \sim \frac{n}{3} . \]

We also have for any graph \( G = (V, \mathcal{E}) \):

\[ (2.6) \quad \sum_{v \in V} d(v) = 2|\mathcal{E}| , \]

Here, \( V \) and \( \mathcal{E} \) are the sets of vertices and edges and \( d(v) \) is the degree of vertex \( v \) in \( G \). Also for any tree \( T = (V, \mathcal{E}) \), the cardinalities of \( V \) and \( \mathcal{E} \) are related by

\[ (2.7) \quad |\mathcal{E}| = |V| - 1 . \]

Thus (2.6) and (2.7) for a binary tree yield

\[ (2.8) \quad |V_1| + 2|V_2| + 3|V_3| = \sum_{v \in V} d(v) = 2(n-1) . \]

The fact that \( V \) is the disjoint union of \( V_1, V_2, V_3 \) yields

\[ (2.9) \quad |V_1| + |V_2| + |V_3| = |V| = n . \]

Taking expectations of (2.9) and (2.8) together with the notation
introduced before the lemma for the expected values, we get

\[(2.10)\quad v_1 + v_2 + v_3 = n,\]

\[(2.11)\quad v_1 + 2v_2 + 3v_3 = 2n - 2,\]

Using (2.5) with (2.10) and (2.11), we get

\[(2.12)\quad n - \frac{(n+1)}{3} - 1 \leq v_2 + v_3 \leq n - \frac{(n+1)}{3},\]

\[(2.13)\quad 2(n-1) - \frac{(n+1)}{3} - 1 \leq 2v_2 + 3v_3 \leq 2(n-1) - \frac{(n+1)}{3},\]

Multiply the right inequality of (2.12) by 3 and write it as

\[(2.14)\quad v_2 + (2v_2 + 3v_3) \leq 2n - 1,\]

Using the left inequality of (2.13) in (2.14), we get

\[(2.15)\quad v_2 + (2(n-1) - \frac{(n+1)}{3} - 1) \leq 2n-1\]

Thus,

\[(2.16)\quad v_2 \leq \frac{(n+1)}{3} + 2.\]

We can similarly show that

\[(2.17)\quad \frac{(n+1)}{3} - 2 \leq v_2.\]

Inequalities (2.16) and (2.17) show that also

\[v_2 \sim \frac{n}{3}.\]

Similarly, we can show that

\[v_3 \sim \frac{n}{3} .\]

Q.E.D.
The result in the theorem above tells us that three kinds of nodes are expected to be equidistributed, and on the average, we do not have subgraphs of the tree which are very long paths.

2.4 The expected level of the leftmost external node

In [17], Ruskey studied the expected level of the leftmost external node under the less realistic assumption that all binary trees are equally likely. He found that this external node is at level 3, on the average. Under our probability measure, namely that all permutations are equally likely, the situation is much different.

Lemma 2.3

The level of the leftmost (rightmost) external node is \( H_n \), the nth harmonic number.

Proof

We shall consider the level of the leftmost internal node. This is the node containing 1 (or the element of the least rank). The level of the leftmost external node is clearly the level of the leftmost internal node plus one.

Let \( E(l_n) \), or simply \( L_n \), be the expected level of the node containing 1 in a tree constructed from a random permutation of \( \{1, 2, \ldots, n\} \). The conditional expectation \( E(l_n) \) when j is in the root, for \( n > 1 \), is given by
(2.18) $E(l_n | j \text{ is in the root}) = \begin{cases} 
0, & j = 1 \\
L_{j-1} + 1, & j > 1, 
\end{cases}$

Notice that $L_1 = 0$ (the tree in this case is a single node containing 1; this node is at level 0). The case $j = 1$ is self explanatory (see Figure 6(a)). The second line of (2.18) means that if $j > 1$ is in the root node, then the node containing 1 will be in the left subtree, and this subtree has $j - 1$ elements, so the expected level of 1 in the original tree is $L_{j-1} + 1$, since the subtrees have the same distribution as the tree itself, and all the levels have to be increased by one when the subtree is attached to the root.

To obtain the unconditional expectation, multiply each conditional expectation by its probability and sum over $j$, thus:

$$L_n = 0 \cdot P(1 \text{ is in the root}) + \sum_{j>1} (L_{j-1} + 1) \cdot P(j \text{ is in the root})$$

Figure 6  Two cases: (a) one in the root  
(b) one in the left subtree

To obtain the unconditional expectation, multiply each conditional expectation by its probability and sum over $j$, thus:
As in the proof of lemma 2.2, to solve this recurrence, write (2.19) with \( n + 1 \) instead of \( n \) and subtract (2.19) from it to get

\[
(n+1)L_{n+1} - nL_n = 1 + L_n .
\]

Equation (2.20) can be written as

\[
L_{n+1} = L_n + \frac{1}{n+1} .
\]

This form is suitable for iterative substitution.

\[
L_n = L_{n-1} + \frac{1}{n} \\
= L_{n-2} + \frac{1}{n-1} + \frac{1}{n} , \\
\vdots \\
= L_1 + \frac{1}{2} + \ldots + \frac{1}{n} , \\
= H_n - 1 .
\]

Q.E.D.

2.5 The expected internal and external path lengths

A measure of the cost of searching the entire tree is the number of comparisons needed to construct the tree from a random permutation or
the number of comparisons needed to find all members of the tree. If we restrict our search queries to keys drawn from the set of elements already stored in the nodes, and if all keys are equally probable, then the sum of the number of comparisons taken over all the elements in this set, often called the internal path length, will be a reasonable measure of the cost of a successful search. If we search for elements not present in the tree, then a good measure would be the sum of the number of comparisons taken over all the external nodes assuming equal likelihood, as usual; i.e., the probabilities that $K$, the key we search for, lies properly between two keys stored in leaf nodes are all equal and are also equal to the probability that $K$ is less (greater) than the element with least (greatest) rank. The following lemmas are easily derivable from several results that may be found in Knuth [1] or Aho, Hopcroft, and Ullman [5]. We give a different proof based on the techniques used in the previous sections just to show their effectiveness in studying trees.

**Lemma 2.4**

For a BST on $n$ nodes

(a) the expected internal path length is $2(n+1)H_n - 4n$.

(b) the expected external path length is $2(n+1)H_n - 2n$.

**Proof**

(a) Denote by $E(p_n | i$ is in the root) the conditional expected value of the internal path length for a tree with $n$ nodes. Then as in
(2.21) \[ E(p_n \mid i \text{ is in the root}) = E(p_{i-1}) + E(p_{n-1}) + n - 1. \]

Relation (2.21) above means that if \( i \) is selected in the root, then the left and right subtrees will have \( i - 1 \) and \( n - 1 \) elements respectively, and each of these \( n - 1 \) elements contributes to the internal path length its level in the subtree it belongs to plus one. Summing over \( i \) and using the simpler notation \( I_n \) instead of \( E(p_n) \), we get

\[
I_n = \sum_{i=1}^{n} P(i \text{ is in the root}) \ast (I_{i-1} + I_{n-1} + n - 1),
\]

(2.22) \[ nI_n = n(n-1) + \sum_{i=1}^{n} (I_{i-1} + I_{n-1}). \]

As in the proofs of the previous lemmas, change indices of the second term in (2.22) and make it identical to the first, then write the resulting equation with \( n + 1 \) instead of \( n \) and subtract to get

(2.23) \[ nI_n - (n + 1)I_{n-1} = 2(n - 1). \]
Define \( I(x) = \sum_{n=0}^{\infty} I_n x^n \), the generating function of the desired sequence of expected internal path lengths.

Multiply (2.23) by \( x^n \) and sum from \( n = 1 \) to \( \infty \).

\[
\sum_{n=1}^{\infty} n I_n x^n - \sum_{n=1}^{\infty} (n + 1) I_{n-1} x^n = \sum_{n=1}^{\infty} 2(n - 1) x^n , \\
= 2x \sum_{n=1}^{\infty} n x^{n-1} - 2 \sum_{n=1}^{\infty} x^n
\]

\[
x \frac{d}{dx} \left( \sum_{n=1}^{\infty} I_n x^n \right) - \frac{d}{dx} \left( x^2 \sum_{n=1}^{\infty} I_{n-1} x^{n-1} \right) = 2x \frac{d}{dx} \left( \sum_{n=1}^{\infty} x^n \right) - \frac{2x}{1 - x}
\]

\[
x \frac{d}{dx}(I(x) - I_0) - 2x I(x) - x^2 I'(x) = \frac{2x}{(1 - x)^2} - \frac{2x}{1 - x}
\]

\[
I'(x) - \frac{2}{1 - x} I(x) = \frac{2x}{(1 - x)^3}.
\]

The general solution of this first order differential equation is

\[
I(x) = \frac{C - 2x + 2 \ln \left( \frac{1}{1 - x} \right)}{(1 - x)^2},
\]

where \( C \) is an arbitrary constant. But \( I'(0) = I_1 = 0 \) implies that \( C = 0 \).

The expansions of \( \ln \left( \frac{1}{1 - x} \right) \) and \( \frac{1}{(1 - x)^2} \) are well known [1]:

\[
I(x) = 2 \left( \sum_{i=0}^{\infty} i x^{i-1} \right) \left( \sum_{m=1}^{\infty} \frac{x^m}{m} - x \right)
\]

hence

\[
I_n = 2 \left( \sum_{m=1}^{n} \frac{n - m + 1}{m} - n \right)
\]
\[ I_n = 2 \left[ (n + 1) \sum_{m=1}^{n} \frac{1}{m} - \sum_{m=1}^{n} (1) - n \right] \]

\[ = 2(n + 1)H_n - 4n. \]

Part (b) could be proved similarly, or more simply using the well-known relation between the external and internal path lengths, \( I \) and \( E \) respectively, namely:

\[ E = I + 2n, \]

take expectations and use part (a).

Q.E.D.
In this chapter, we find the recurrence relations governing the length of the shortest and longest paths from the root to an external node in the extended BST, introduced in chapter I, and use them to establish computational formulas for the average values of these path lengths for trees in this class, for any finite tree size $n$. In the next chapter, we shall perform asymptotic analysis of the shortest and longest paths from the root to an external node in an extended BST, as $n \to \infty$.

Clearly the height of an extended BST is bounded below by $\lceil \log_2(n + 1) \rceil$, but could degenerate into a linear list (or even worse, height $= n$
shortest path $= 1$

height $= \lceil \log_2(n + 1) \rceil$
shortest path $= \lfloor \log_2(n + 1) \rfloor$

Figure 8  Trees with extremal path lengths
by a constant factor, since the space needed is larger than a linked list as each node has two pointers); see figure 8. So we address the questions: what is the probability of getting the minimum height tree, and what are the average height and shortest path length of a tree built from a random permutation?

3.1 Permutations and the height of a BST

Suppose we have a random permutation of the integers 1, 2, ..., n as an input sequence from which we construct a BST. It is clear that the choice of the root (the first element of the permutation) determines the elements of the left and right subtrees, and this applies recursively for the subtrees. Furthermore, the arrangement of the elements that will occupy the right subtree with respect to those that will occupy the left subtree is immaterial. What really matters is the arrangement of the elements of each subtree among themselves. For example, the two permutations

and

Figure 9 Two permutations that give the same BST

will give the same extended binary tree, illustrated in figure 10.

Figure 10 The BST corresponding to the above permutations
The two permutations are structurally the same because 4 was chosen as the root, and the integers 2, 1, 3 (the squares in figure 9) appear in the same order (left to right) in both permutations regardless of the intervening numbers (which belong to the right subtree). This rule will apply recursively in the subtrees, e.g., in the left subtree, the left most integer is 2, which appears first, so it will be the root of the left subtree, etc. This intuition suggests the following definition and lemmas.

Definition

Let X and Y be two disjoint sets of sizes m and n respectively. Define \( A_X \) and \( A_Y \) to be the number of "legal" permutations of these two sets, respectively. Define an arrangement of the set \( X \cup Y \) as legal if the elements of X preserve legality of arrangement among themselves, and so do the elements of Y.

Another way to look at this definition is to consider that if we have an arrangement of \( X \cup Y \) which is legal, then if we take out all the elements of X, one of the \( A_Y \) legal ways of arranging Y is left, and vice versa.

Lemma 3.1

The number of legal arrangements \( A_X \cup Y \) is

\[
A_X \cup Y = A_X A_Y \frac{(n + m)!}{n! m!} = A_X A_Y \binom{n + m}{n}.
\]
Proof

Without loss of generality, we assume that $m \geq n$. Let $X_1, X_2, \ldots, X_m$ be a legal arrangement of $X$.

Assume that we also have a legal arrangement of $Y$. We can partition it into $k$ divisions, $1 \leq k \leq n$ (a division may contain more than one element). Each of these $k$ divisions can fit into one place between the $X$s in the above legal arrangement of $X$ to give a legal arrangement of $XU^Y$ (see figure 11 above). The number of ways to fit the $k$ divisions is $\binom{m+2}{k}$, since we may select any $k$ of the $m+1$ "holes" between the $X$s, considering the places to the left and right of all $X$s as holes. But these $k$ divisions of a permutation of $Y$ can be obtained in one of $\binom{n-1}{k-1}$ ways (these are obtainable by putting $k-1$ vertical bars between the $Y$s, and there are $n-1$ gaps between the elements of $Y$). Hence, for one choice of the legal arrangements of $X$ and one choice of the legal arrangements of $Y$, we have $\binom{m+1}{k}\binom{n-1}{k-1}$ ways to intersperse the elements of $Y$ between the elements of $X$ to get one of the $A^X_{U^Y}$ arrangements, if we break up a legal arrangement of $Y$ into $k$ divisions. But $k$ can range from 1 to $n$, and we have $A^X_X$ and $A^Y_Y$ arrangements of $X$ and $Y$, respectively. Therefore:
\[ A_{X \cup Y} = \sum_{k=1}^{n} A_X A_Y \left( \begin{array}{c} m + 1 \\ k \\ \end{array} \right) \left( \begin{array}{c} n - 1 \\ k - 1 \end{array} \right) = A_X A_Y \sum_{k=1}^{n} \left( \begin{array}{c} m + 1 \\ k \\ \end{array} \right) \left( \begin{array}{c} n - 1 \\ k - 1 \end{array} \right). \]

The summation is a well known one [1], and it simplifies to
\[ A_{X \cup Y} = A_X A_Y \left( \begin{array}{c} n + m \\ m \end{array} \right). \]
Q.E.D.

**Example**

If all arrangements of X are legal, then \( A_X = m! \), and if all
arrangements of Y are legal, then \( A_Y = n! \), in which case
\[ A_{X \cup Y} = m! \left( \begin{array}{c} n + m \\ n \end{array} \right) = (n + m)! , \]
i.e., all permutations of \( X \cup Y \) (which has cardinality \( n + m \)) are legal.

This lemma enables us to study a special subclass of BSTs.

**3.2 Perfect trees**

**Definition**

We call a tree perfect if it includes \( 2^m - 1 \) internal nodes (m is
an integer) and level \( i \) has \( 2^i \) elements exactly, for \( i = 0, 1, \ldots, m - 1 \).

Such a tree has the shape of a triangle, and the height of the extended
perfect tree is \( m \).

Let \( T(n) \) be the number of permutations of \{1, 2, \ldots, n\} that give
rise to a perfect binary search tree (\( n \) has to be of the form \( 2^m - 1 \)).
In order to get a perfect tree, the integer $\frac{n+1}{2}$ has to be the first element in the permutation as in figure 12.

![Figure 12](image)

Figure 12 A perfect tree on n nodes ($n = 2^m - 1$)

The elements that will occupy the left subtree are the integers

$$L = \{1, 2, \ldots, \frac{n+1}{2} - 1\},$$

and those of the right subtree are

$$R = \{\frac{n+1}{2} + 1, \frac{n+1}{2} + 2, \ldots, n\}.$$

These two sets are disjoint, and to get a perfect tree, we only need to have perfect right and left subtrees. The way they are interspersed is unimportant, thus the number of "legal" arrangements of $R$ is $T(\frac{n-1}{2})$ and the number of "legal" arrangements of $L$ is $T(\frac{n-1}{2})$. Using the above lemma:

$$T(n) = T(\frac{n-1}{2})T(\frac{n-1}{2})^{\left(\frac{n-1}{2}\right)} ,$$

$$T(n) = \frac{(n-1)!}{\left[\frac{(n-1)!}{2}\right]} * T^2(\frac{n-1}{2}), T(1) = 1.$$
Repeated substitution leads to

\[ T(n) = \frac{(n - 1)!}{\left(\frac{n - 1}{2}\right)!^2} \frac{\left(\frac{n - 1}{2} - 1\right)!^2}{\left(\frac{n - 1}{2} - 1\right)!^4} \left(\frac{n - 1}{2} - 1\right) \left(\frac{n}{2} - 1\right) \]

= \frac{(n - 1)!}{(\frac{n - 1}{2})^2} \frac{\tau_4(n - 3)}{[(\frac{n - 3}{4})!]^4}

= \frac{(n - 1)!}{(\frac{n - 1}{2})^2(\frac{n - 3}{4})^4} \frac{\tau_8(n - 7)}{[(\frac{n - 7}{8})!]^8}

\cdot \cdot \cdot 

= \prod_{i=1}^{m-1} \frac{(n - 1)!}{(\frac{n - (2^i - 1)}{2^i})^2} \frac{\tau_2^{(m-1)}(n - (2^m - 1))}{2^{(m-1)}}

But \( n = 2^m - 1 \), and \( T(1) = 1 \), thus the solution of the above recurrence is

\[ T(n) = \frac{(n - 1)!}{\prod_{i=1}^{m-1} (2^{m-1} - 1)^2}, \quad m = \log_2(n + 1). \]

Hence, the probability of building a perfect tree from a random permutation of the integers 1, 2, ..., \( 2^m - 1 \) is \( \frac{T(n)}{n!} \) (since \( n! \) is the total number of permutations), i.e.,

\[ \frac{1}{n \prod_{i=1}^{m-1} (2^{m-1} - 1)^2}. \]
3.3 Permutations and trees of certain path lengths

For the general case, $n \neq 2^m - 1$ for any $m$. But we can try to compute the expected height as follows. Let $T(n, k)$ be the number of permutations on $n$ integers that give binary search trees of height at most $k$. But again, we can have such a permutation if $i$ is chosen at the root and both the left and right subtrees are of height $k - 1$ at most. The left subtree is constructed from the set $L = \{1, 2, \ldots, i - 1\}$ and the right subtree is constructed from $R = \{i + 1, i + 2, \ldots, n\}$; (see figure 13).

![Figure 13](image)

Figure 13  
i is in the root and the left and right subtrees are constructed from the sets $L$ and $R$

Thus $L$ can be permuted in $T(i - 1, k - 1)$ ways and $R$ in $T(n - i, k - 1)$ ways to give subtrees of height at most $k - 1$, and these arrangements and be interspersed in $\sum_{i=1}^{n-1} T(n - i, k - 1) T(i - 1, k - 1)$ ways according to the lemma. And $i$ can be chosen to be any integer in $\{1, \ldots, n\}$, hence

\[
T(n, k) = \sum_{i=1}^{n} \left( \begin{array}{c} n - 1 \\ i - 1 \end{array} \right) T(n - i, k - 1) T(i - 1, k - 1),
\]

with $T(n, 0) = 0$ for $n > 0$ and $T(0, k) = 1$ for all $k \geq 0$. 

The number of permutations that give trees of height exactly \( k \) is \( T(n, k) - T(n, k - 1) \); hence, the expected height is

\[
d_n = \sum_{k=0}^{\infty} k \cdot (\text{probability of getting a permutation that gives a tree of height } k)
\]

\[
= \sum_{k=0}^{\infty} k \cdot \frac{T(n, k) - T(n, k - 1)}{n!}.
\]

This sum is not really infinite since the probability of getting a tree of depth \( m, m > n \), is zero; thus

\[
d_n = \sum_{k=0}^{n} \frac{T(n, k) - T(n, k - 1)}{n!}.
\]

The above relation is a handy formula to compute the expected depth of a binary search tree. Appendix A gives a table of \( d_n \) for some smaller values of \( n \); also the table gives the values \( \ln n \), the ratio \( \frac{d_n}{\ln n} \), and the variance of \( d_n \). These results support our asymptotic estimates of the next chapter (we cannot consider this a verification, as we can never verify an asymptotic).

Similar results concerning the shortest path could be obtained had we started with \( t(n, k) \), the number of permutations on \( n \) integers that give a BST of shortest path at least \( k \). Arguing exactly along the same line to establish recurrence (3.1), we get

\[
(3.2) \quad t(n, k) = \sum_{i=1}^{n} \binom{n-1}{i-1} t(n-i, k-1) t(i-1, k-1).
\]
The initial conditions are $t(n, 0) = n!$ for $n \geq 0$ and $t(0, k) = 0$ for all $k > 0$.

$S_n$, the average shortest path length for BSTs on $n$ nodes, will be given by

$$S_n = \sum_{k=0}^{n} \frac{t(n, k) - t(n, k + 1)}{n!}.$$ 

Appendix B gives a table of $S_n$ computed this way and in $n$, $\frac{S_n}{\ln n}$, and the variance of $S_n$.

3.4 The exponential generating function of the sequences $T(n, k)$ and $t(n, k)$

The binomial coefficients in the right hand sides of recurrences (3.1) and (3.2) suggest the use of exponential generating functions, instead of ordinary ones, in solving these recurrences.

Starting with $T(n, k) = \sum_{i=1}^{n} \binom{n-1}{i-1} T(n-i, k-1) T(i-1, k-1)$, first write it as

$$T(n, k) = \sum_{i=1}^{n} \frac{T(n-i, k-1) \ast T(i-1, k-1)}{(n-1)! \ast (n-i)! \ast (i-1)!},$$

then multiply both sides by $z^{n-1}$ and sum over $n$, writing $z^{n-1}$ as $z^{n-1} \ast z^{i-1}$ on the right hand side.
We get

\[ \sum_{n=1}^{\infty} \sum_{i=1}^{n} \frac{T(n, k)}{(n-1)!} z^{n-1} = \]

\[ \sum_{n=1}^{\infty} \sum_{i=1}^{n} \frac{T(n-1, k-1)}{(n-1)!} z^{n-1} \times \frac{T(i-1, k-1)}{(1-1)!} z^{i-1}. \]

Define \( P_k(z) = \sum_{n=0}^{\infty} \frac{T(n, k)}{n!} z^n \) to be the exponential generating function for the numbers \( T(n, k) \).

Now \( \frac{dP_k(z)}{dz} = \sum_{n=1}^{\infty} \frac{T(n, k)}{(n-1)!} z^{n-1} \), so in the ring of formal power series, (3.3) becomes

\[ (3.4) \quad P_k'(z) = P_{k-1}(z). \]

The initial conditions \( T(n, 0) = 0 \) and \( T(0, 0) = 1 \) imply that

\[ P_0(z) = T(0, 0) + 0 + 0 + \ldots = 1. \]

Also, \( T(0, k) = 1, k \geq 0 \), implies that

\[ P_k(0) = T(0, k) + 0 + 0 + \ldots = 1, \text{ for } k \geq 0. \]

The generating function \( p_k(z) \) of the sequence \( \frac{t(n, k)}{n!} \)
is defined similarly, and it is clear that

\[ (3.5) \quad p_k'(z) = p_{k-1}(z) \]
since the recurrence relations for \( t(n, k) \) and \( T(n, k) \) are identical.
The only difference will be the initial conditions.

The condition \( t(n, 0) = n! \) implies that

\[
p_0(z) = \sum_{n=0}^{\infty} \frac{t(n, 0)}{n!} z^n
\]

\[
= \sum_{n=0}^{\infty} z^n
\]

\[
= \frac{1}{1 - z}.
\]

Also, \( t(0, k) = 0 \) for \( k > 0 \) means that

\[
p_k(0) = t(0, k) = 0.
\]

The explicit solutions of (3.4) and (3.5) are not available, perhaps because this class of trees is not "simple" in the sense explained in the next chapter. Flajolet and Odlyzko [12] encountered similar difficulties determining the number of binary trees with \( n \) nodes and height \( h \).
CHAPTER IV
THE SHORTEST AND LONGEST PATHS IN A BST:
AN ASYMPOTOTIC ANALYSIS

The asymptotic value of the length of the longest path from the root to a leaf node has been studied for several classes of trees [12 - 15], [17]. The shortest path does not seem to have had much attention, although it is the other end of the spectrum, and it represents the best case performance of some tree algorithms.

Knuth [1] studied the average length of the path from the root to a randomly selected external node for the class BST. He found that this is $\sim 2\ln n$. Robson [15] found bounds on the longest path length for the same class. Ruskey [17] studied the expected lengths of paths from the root to all external nodes for the class of binary trees. The average height of a binary tree when all binary trees are equally likely was found by Flajolet and Odlyzko [12]. They showed that this height is $\sim 2\sqrt{n\ln n}$. In fact, in [12], they proved a more general theorem. They showed that when all n-node trees are equally likely in a "simple" family of trees $S$, the average height is equivalent to $c(S)\sqrt{n}$, where $c(S)$ is a constant depending on the nature of the family $S$. Meir and Moon [18] defined a family of trees to be simple if the generating function

$$y = y(x) = \sum_{n=1}^{\infty} y_n x^n$$
of the sequence \( y_n \), the number of trees with \( n \) nodes, takes the form

\[
y = x \theta(y),
\]

where \( \theta(y) \) is a power series in \( y = y(x) \) with non-negative coefficients.

It turns out that \( y_n \) has a straightforward interpretation in terms of the coefficients of the power series \( \theta(y) \) for all simply generated families. (It was shown in [18] that all the families of table 1 are simply generated.) But the number of differently labeled BSTs with \( n \) nodes is \( n! \) (one for each \( n \)-long permutation). Thus, for this family

\[
y = \sum_{n=1}^{\infty} n! x^n,
\]

hence it is not simple.

In studying the height of random trees, the authors [12 - 15], [17] used a variety of methods. Often they started with explicit recurrence relations which were not easy to solve, then they appealed to some clever methods to find asymptotic estimates. For example, De Bruijn, Knuth, and Rice [13] used the Mellin integral transform, and this method has been the starting point for some analyses of digital search trees and Patricia trees. Flajolet and Odlyzko [12] proposed a fairly general technique. Their idea is to study the locations of the singularities of the generating function, which are closely related to the coefficients of the generating function.

Among other relevant results, we should mention Stepanov [19] who established the limiting distribution of the height of the un-ordered random tree. For a very comprehensive survey of numerous other results in this area, see Moon [20].
The class of binary search trees is a completely different story. It is not simple, as mentioned above. In the previous chapter, we obtained computational formulas for the height and shortest path length of a BST from recurrence relations. The rate of convergence of the averages of these random variables divided by \( \ln n \) is very slow and gets slower and slower as \( n \) increases, as evident from the tables in appendices A and B. The main difficulty posed by the direct combinatorial approach based on counting arguments in chapter III was that the recurrence relation was unwieldy. In this chapter, we shall try to overcome this hurdle in the asymptotic analysis. We employ a very simple idea. Draw a horizontal line in the tree at level \( k(n) \) and seek a choice of \( k \) that makes the number of leaves vanish (for large \( n \)) below (above) this line; this will be a bound on the longest (shortest) path length. The apparent convergence of \( \frac{S_n}{\ln n} \) and \( \frac{d_n}{\ln n} \) suggests the choice \( k(n) = c \ln n \), where \( c \) is some constant to be found later.

Two results are most relevant to the results presented below. Robson [15] drew a similar conclusion about the longest path, using an entirely different approach. He considered the probability \( P(d) \) that a particular input sequence of \( d - 1 \) directions (branch left or right) leads from the root to a terminal node. From \( P(d) \), he obtained an upper bound on the probability that the tree has height exactly \( d \), then he analyzed the resulting function.

Using the idea of percolation due to Firsch and Hammersley [21], Pittel [22] proved the existence of two absolute constants \( \beta_1, \beta_2 \), such that the limiting value as \( n \to \infty \), of the shortest and longest paths are equal to \( \beta_1 \ln n \) and \( \beta_2 \ln n \) with probability equal to one (this was
4.1 Preliminaries and results

Definition

Define $h_n = h(w^{(n)})$ and $H_n = H(w^{(n)})$, respectively, as the lengths of the shortest and longest paths in $T_n$ from the root to an external node. Introduce also $b_n = b(w^{(n)}) = H(w^{(n)})/h(w^{(n)})$, which serves as a natural measure of how poorly balanced the tree $T_n$ is.

It is obvious that

\[
\begin{align*}
1 & \leq h(w^{(n)}) \leq \lceil \log_2(n + 1) \rceil, \\
\lceil \log_2(n + 1) \rceil & \leq H(w^{(n)}) \leq n, \\
1 & \leq b(w^{(n)}) \leq n.
\end{align*}
\]

Thus these characteristics may vary widely from one permutation to another.

Our goal is to study asymptotic properties, as $n \to \infty$, of the typical values of $h_n$, $H_n$, $b_n$ under the assumption that all $n!$ permutations are equally probable.

In this chapter, we prove

Theorem 4.1

For each $\varepsilon > 0$,

\[
P(h_n/\ln n \geq \alpha_1 - \varepsilon \text{ and } H_n/\ln n \leq \alpha_2 + \varepsilon + 1, \text{ as } n \to \infty;
\]

here $\alpha_1$ and $\alpha_2$ are the two roots of the equation

\[
\alpha + \alpha \ln(2/\alpha) = 1, \quad \alpha_1 \approx 0.37, \quad \alpha_2 \approx 4.31.
\]
NOTE: According to this theorem, if \( n \) is large, then with high probability all the paths of the tree \( T_n \) are of the same logarithmic order, as in the case of the best balanced tree, see (4.1). More precisely, the balance measure is asymptotically almost always less than \( \left( \frac{\alpha_2}{\alpha_1} \right) + \epsilon = 11.65 + \epsilon, \forall \epsilon > 0. \)

Intermediate estimates which we obtain while proving this theorem enable us to get a stronger version of (4.2). To formulate the corresponding statement, we need to introduce the notion of an infinite random permutation. To this end, notice first that the components of the sequential ranks vector \( r^{(n)}_n = (r_1, \ldots, r_n) \) of the \( n \)-long random permutation \( w^{(n)}_n \) satisfy the conditions: (a) \( r_j \) is uniformly distributed on \( R_j = \{1, \ldots, j\} \), and (b) \( r_1, \ldots, r_n \) are independent. So, slightly abusing notations, let us consider the space of infinite integer-valued sequences \( w = (r_1, r_2, \ldots) \), \( r_j \in R_j \), and the probability measure on it induced by the conditions (a) \( r_j \) is uniformly distributed on \( R_j = \{1, \ldots, j\} \), and (b') that \( r_1, r_2, \ldots \) are independent.

Given any such sequence \( w \), our algorithm constructs an infinite sequence \( \{t(w^{(n)}_n)\}_{n=1}^{\infty} \) of nested binary search trees, (here \( w^{(n)}_n = (r_1, \ldots, r_n) \)), which can be naturally considered as the ever-increasing proper subgraphs of the corresponding infinite binary tree \( t(w) \). Hence, we come to \( \{h_n, h_n, b_n\}_{n=1}^{\infty} \), all determined on the same probability space, and it becomes natural to study the asymptotic behavior of these sequences.

This model corresponds to constructing a BST by reading consecutively a stream of input records, where the sequential rank of each new coming record assumes all its feasible values with equal probabilities independently of the sequential ranks of the records read thus far.
We will also prove

Theorem 4.2

With probability one,

\[ \lim \inf_{n} \frac{H_{n}}{\ln n} \geq \alpha_1, \quad \lim \sup_{n} \frac{H_{n}}{\ln n} \leq \alpha_2, \quad (n \to \infty). \]

This result reveals that almost surely there exists \( n(w) \) such that all the trees \( T(w^{(n)}) \) with \( n \geq n(w) \) are well balanced (i.e., all paths from the root to a leaf node have length \( \Theta(\log n) \)).

4.2 Auxiliary combinatorial relations

Notation

Let \( X_{nk} \) be the random number of external nodes of \( T_n \) at distance \( k \) from the root; by definition, \( X_{n0} = \delta_{n0} \). Denote \( E_{nk} = E(X_{nk}) \) the expected value of \( X_{nk} \).

Lemma 4.1

For \( n \geq 1, k \geq 1 \),

\[ E_{nk} = \frac{2}{n} \sum_{i=0}^{n-1} E_{i,k-1}. \]

Proof

Let \( t_{n}^{(1)}, t_{n}^{(2)} \) be the left and right subtrees of \( t_n \), and denote \( T_{n}^{(1)}, T_{n}^{(2)} \) their corresponding extensions. (We assume that the extension
of the empty tree consists of a single square node.) Denote by $U_n^{(s)}$ the number of internal nodes of $T_n^{(s)}$, and $V_n^{(s)}$ the number of external nodes which are at distance $k - 1$ from its root, $(s = 1, 2)$. Clearly

$$U_n^{(1)} + U_n^{(2)} = n - 1, \quad V_{nk} = V_{n,k-1}^{(1)} + V_{n,k-1}^{(2)}$$

According to the BST-algorithm and probabilistic properties of $(r_j)_{j=1}^\infty$, we have: conditioned on the event $(w: U_n^{(1)} = 1)$, $T_n^{(1)}$ and $T_n^{(2)}$ are distributed as $T_{i-1}$ and $T_{n-1-1}$, respectively, $(0 \leq i \leq n - 1)$. Hence

$$E(X_{nk} | U_n^{(1)} = 1) = E(V_{n,k-1}^{(1)} + V_{n,k-1}^{(2)} | U_n^{(1)} = 1) = E(X_{i,k-1}^{(1)} + E(X_{n-1-1,k-1}^{(2)} = E_{i,k-1}^{(1)} + E_{n-1-1,k-1}^{(2)}$$

Since $P(U_n^{(1)} = 1) = \frac{1}{n}$, $0 \leq i \leq n - 1$,

We obtain then

$$E_{nk} = \frac{1}{n} \sum_{i=0}^{n-1} E(X_{nk} | U_n^{(1)} = 1) = \frac{\left[ E_{i,k-1}^{(1)} + E_{n-1-1,k-1}^{(2)} \right]}{n} = \frac{1}{n} \sum_{i=0}^{n-1} E_{i,k-1}^{(1)}$$

To formulate the next lemmas, we need the signless Stirling numbers $S(v, u)$ of the first kind, [24]. They are defined for $0 \leq u \leq v$ by conditions:

$$S(v, 0) = \delta_{v0}, \quad (4.6)$$
and for $1 \leq \mu \leq \nu$

\[
\sum_{\mu=1}^{\nu} x^\mu S(\nu, \mu) = x(x + 1) \ldots (x + \nu - 1) = \langle x \rangle_\nu .
\]

$S(\nu, \mu)$ are also uniquely determined by the recurrence relation

\[
S(\nu + 1, \mu + 1) = S(\nu, \mu) + \nu S(\nu, \mu + 1), \quad 0 \leq \mu \leq \nu,
\]

and the boundary conditions (4.6).

**Lemma 4.2**

For $n \geq k \geq 0$,

\[
E_{nk} = 2^k S(n, k)/n! .
\]

**Proof**

Let $m \geq 2$; comparing (4.5) when $n = m$ and $n = m - 1$ leads to

\[
mE_{mk} - (m - 1)E_{m-1,k} = 2E_{m-1,k-1} .
\]

Introducing $\bar{S}(m, k) = 2^{-k} E_{mk} m!$, we transform (2.6) to

\[
\bar{S}(m, k) = \bar{S}(m - 1, k - 1) + (m - 1) \bar{S}(m - 1, k).
\]

By definition of $\bar{S}(1, 1)$ it is clear that

\[
\bar{S}(m, 0) = \delta_{m0} .
\]

Comparing (4.11), (4.12) with (4.8), (4.6) yields (4.9).
Note According to lemma 4.2 and (4.7),
\[
\begin{align*}
E(\sum_{k=1}^{n} X_{nk}) &= \frac{\langle 2 \rangle}{n!} = n + 1, \\
E(\sum_{k=1}^{n} X_{nk} 2^{-k}) &= \frac{\langle 1 \rangle}{n!} = 1.
\end{align*}
\]
These relations should be anticipated because \( \sum_{k=1}^{n} X_{nk} \) is just the total number of square nodes, which is always equal to \( n + 1 \), and \( \sum_{k=1}^{n} X_{nk} 2^{-k} \) is known to equal one for each extended binary tree [1].

Also, lemma 4.2 can be proved differently. We can solve recurrence (4.5) using generating functions. This will be done when we generalize the problem in m-way trees in chapter VI.

4.3 Asymptotic estimates of \( E_{nk} \)

In what follows, we shall assume that \( k, n \to \infty \) in such a way that \( \alpha = k/\ln n \) is bounded from above.

Lemma 4.3

If \( \alpha \) is bounded away from 0, then
\[
E_{nk} = \frac{\rho(\alpha)}{\sqrt{\ln n}} \exp \left[ g(\alpha) \ln n \right] (1 + o(1)),
\]
where
\[
\begin{align*}
g(\alpha) &= \alpha + \alpha \ln(2/\alpha) - 1, \\
\rho(\alpha) &= (\sqrt{2\pi \alpha} r(\alpha))^{-1} \exp (\alpha - 1),
\end{align*}
\]
and \( r(\cdot) \) denotes the gamma function.
Lemma 4.4

\begin{align}
(4.16) \quad & \sum_{j=1}^{k} E_{n j} \leq c(1 - \alpha/2)^{-1} \exp \{g(\alpha) \ln n\}, \text{ if } \alpha < 2, \\
(4.17) \quad & \sum_{j=k}^{n} E_{n j} \leq c(1 - 2/\alpha)^{-1} \exp \{g(\alpha) \ln n\}, \text{ if } \alpha > 2,
\end{align}

and in both cases, \( c \) depends on \( \alpha \).

Proof of lemmas 4.3, 4.4

(1) As \( E_{n k} = 2^k S(n, k)/n! \), we need an estimate of the Stirling number \( S(n, k) \). According to a (more general) result due to Moser and Wyman [25],

\[ S(n, k) \leq k^{n-\frac{1}{2}} \left( \frac{n}{\pi} \right)^{\frac{1}{2}} \exp \left( -\frac{1}{2} \left( \frac{n}{\pi} \right)^{\frac{1}{2}} \right) r(n, r) \frac{r^{n-\frac{1}{2}}}{\sqrt{2\pi k}} r(r) (1 + o(1)), \]

where \( r \) is the only positive root of the equation

\[ \sum_{j=0}^{n-1} \frac{r}{(r + j)} = k, \]

and

\[ n - 1 H = k - \sum_{j=0}^{n-1} \frac{r^2}{(r + j)^2}. \]

It can be seen that under the assumption concerning \( k, [1], \)

\[ r = \alpha + \alpha \psi(\alpha)/\ln n + 0(1/\ln^2 n) \]

\[ H = k(1 + o(1/\ln n)) \]

where \( \psi(\alpha) = r'(\alpha)/r(\alpha) \).
Hence
\[ S(n, k) = \frac{r(n + r)(1 + o(1))}{\alpha^{1/2} \pi^k r(\alpha) \exp(\alpha \psi(\alpha))}. \]

Also, in view of the Stirling formula
\[ r(z + 1) = \sqrt{2\pi z} \left(\frac{z}{e}\right)^z (1 + o(1)), \quad z \to \infty, \]
we obtain
\[ r(n + r)/n! = r(n + r)/r(n + 1) \]
\[ = \exp[(r - 1) * (\ln n + 1) + O(1/n)] \]
\[ = \exp[\ln n (\alpha - 1) + \alpha - 1 + \alpha \psi(\alpha) + O(1/\ln n)]. \]

A combination of \( E_{nk} = 2^k S(n, k)/n! \), (4.18), and (4.19) yields (4.13).

(2) For brevity, we shall prove only (4.16). By (4.7) and lemma 4.2,
\[ E_{nj} = (2^j/n!) \text{coeff}_{x^n} x^j \leq (2^j/n!) <x^n_x^n>/x^j \]
for each positive \( x \). If \( x > 2 \), then
\[ \sum_{j=k}^{n} E_{nj} \leq \frac{(2/x)^j}{(1 - 2/x)} \]
\[ = \frac{[2^{r(n + n)/r(n + 1)}]}{r(x) (1 - 2/x)}. \]
Here, again by the Stirling formula,

\[
(2/x)^k \Gamma(x + n)/\Gamma(n + 1)
= \exp[k \ln (2/x) + (x - 1) \ln n + O(1)]
= \exp[k \ln n (\alpha \ln (2/x) + x - 1) + O(1)],
\]

and, choosing \( x = \alpha \) we come to (4.16). (\( \alpha \) is the point where \( \alpha \ln (2/x) + x - 1 \) assumes its minimum, which explains why \( x \) is chosen this way.)

### 4.4 Proof of theorems 4.1, 4.2

An elementary study of \( g(\alpha) \) shows that it is convex upward, achieves its maximum value 1 at \( \alpha = 2 \), equals 0 at \( \alpha_1 \approx 0.37 \) and \( \alpha_2 \approx 4.31^+ \), and is negative everywhere outside \([\alpha_1, \alpha_2] \).

![Figure 14 The graph of the function g(\alpha)](image)

\[\text{Figure 14 The graph of the function } g(\alpha)\]

*Robson [15] found this number using a different approach. See the introduction of this chapter.*
(a) Given \( \epsilon > 0 \), introduce \( k = \lfloor (\alpha_2 + \epsilon) \ln n \rfloor \). By lemma 4.4:

\[
P(H_n \geq (\alpha_2 + \epsilon) \ln n) = P(\bigcup_{j=k}^{n} (X_{n,j} > 0))
\]

\[
\leq \sum_{j=k}^{n} P(X_{n,j} > 0) \leq \sum_{j=k}^{n} E(X_{n,j}) = \sum_{j=k}^{n} E_n j
\]

\[
\leq c \exp \left[ g(\alpha_n) \ln n \right],
\]

where

\[
\alpha_n = \alpha_2 + \epsilon + O(1/\ln n) = \alpha_2 + \epsilon.
\]

Hence

\[
(4.20) \quad P(H_n \geq (\alpha_2 + \epsilon) \ln n) \leq c n^{-c_1}, \quad c_1 = c_1(\epsilon) > 0.
\]

Similarly

\[
(4.21) \quad P(H_n \leq (\alpha_1 - \epsilon) \ln n) \leq c n^{-c_2}, \quad c_2 = c_2(\epsilon) > 0.
\]

(4.20), (4.21) prove theorem 4.1.

(b) Let us show that

\[
(4.22) \quad P(\limsup_{n} H_n/\ln n \leq \alpha_2) = 1;
\]

in other words that for each \( \alpha > \alpha_2 \)

\[
P(H_n \geq \alpha \ln n \text{ infinitely often}) = 0.
\]

To this end, it would suffice to show (Borel-Cantelli lemma [26]) that

\[
(4.23) \quad \sum_{n=1}^{\infty} P(H_n \geq \alpha \ln n) < + \infty.
\]

But (4.23) does not follow from (4.20) as \( c_1 \) depends on \( \epsilon \), and in fact goes to 0 as \( \epsilon \) gets smaller. Still, a simple idea (see also [27],
[28], [29]) based on the observation that $H_n$ increases with $n$, and $\ln n$ is slowly varying, helps to overcome this obstacle.

Choose an integer $k$ so large that $kc > 1$. Then

$$\sum_{m=1}^{\infty} P(H_k \geq \alpha \ln(m^k)) \leq c \sum_{m=1}^{\infty} m^{-kc} < \infty,$$

so that (Borel-Cantelli; after all)

$$P(\lim sup_{m} H_k/\ln(m^k) \leq \alpha_2) = 1.$$ 

Further, given $n$, let $m(n)$ be determined by

$$m^k(n) \leq n < (m(n) + 1)^k;$$

clearly, $m(n) \to \infty$ as $n \to \infty$. Since

$$\frac{H_n}{\ln n} \leq \frac{H}{(m(n)+1)^k/\ln((m(n) + 1)^k)} (1 + O(1/(m(n) \ln m(n))),$$

we have that

$$\lim sup_{n} \frac{H_n}{\ln n} \leq \alpha_2$$

with probability one, too.

The case of $h_n$ can be considered in a similar way. Theorem 4.2 is thus proven.

4.5 A remark on the distribution of endpoints

The estimate of lemma 4.2 can be used to obtain a result a little stronger than Knuth's estimate of the average path length. Let $N_n(c)$
be the random total number of external nodes, such that their distance from the root lies outside the interval

$$[(2 - c) \ln n, (2 + c) \ln n].$$

**Lemma 4.5**

$$\forall \epsilon > 0, \quad P(N_n(\epsilon) \geq cn^{1-\delta/2}) = o(n^{-\delta/2}), \quad \delta \in (0,1).$$

**Proof**

By (4.16) and (4.17), for all $\epsilon > 0$,

$$E(N_n(\epsilon)) \leq c[\max(g(2 - c), g(2 + c)) \ln n]$$

$$\leq cn^{1-\delta}, \quad \delta = \delta(\epsilon) \in (0,1),$$

as $\max g(\alpha) = 1$ is achieved at $\alpha = 2$.

Hence, by Markov's inequality,

$$P(N_n(\epsilon) \geq cn^{1-\delta/2}) \leq \frac{E(N_n(\epsilon))}{cn^{1-\delta/2}} = o(n^{-\delta/2}).$$

Thus the limit of this probability approaches zero as $n \to \infty$.

Q.E.D.

This result shows that not only the average expected path length from root to a randomly selected external node is $-2 \ln n$ as Knuth [1] has found, but also almost the entire collection of endpoints is concentrated around this level, with high probability, indicating a small variance.
CHAPTER V
m-WAY TREES AND THEIR BALANCING

5.1 m-way search trees defined: an insertion model

The binary search tree model of the previous chapters has a very natural generalization: the m-way search tree. In this chapter, we consider this model, and we discuss two balancing techniques. In the next chapter, we shall perform an asymptotic analysis on some of the properties of this class of trees.

Here we shall start with the notion of the infinite random permutation. Consider the sequence \( w = (w(1), w(2), \ldots) \) of distinct numbers. Again, one may think of \( w \) as a stream of input records for a computer. The computer reads the records in turn and constructs a sequence of nested trees \( \{t_n\}_{n=0}^\infty \) such that if \( w^{(n)} \) is the \( n \)-long initial segment of \( w \), then \( t_n \) is an m-way search tree defined recursively essentially as in [4]. Namely, an m-way search tree on \( n \) elements \( t_n \) is either empty, or each node contains at most \( m - 1 \) elements (keys) \( K_1, K_2, \ldots, K_i, 1 \leq i \leq m - 1 \), sorted left to right \( (K_1 < K_2 < \ldots < K_i) \). Furthermore, if \( n < m \), then all the elements are in the root, and if \( n \geq m \), then the first \( m - 1 \) elements are in the root, and the remaining \( n - m + 1 \) elements are distributed among subtrees \( T_{n1}, \ldots, T_{nm} \) subject to

(i) the elements of \( T_{n1} \) are less than all the elements in
the root;

(iii) the elements of $T_{n1}$, $1 < i < m$, lie properly between keys $K_{i-1}$ and $K_i$ of the root;

(iii) the elements of $T_{nm}$ are greater than all the elements of the root.

(iv) the subtrees $T_{n1}, \ldots, T_{nm}$ are also $m$-way search trees.

The following example demonstrates the construction algorithm informally.

Example

Consider the permutation

$$w^{(7)} = (4, 2, 3, 5, 7, 6, 1)$$

Let us construct a ternary tree ($m = 3$). The permutation $w^{(7)}$ gives the following sequence of trees:
Figure 15  The sequence of ternary trees constructed from the permutation (4, 2, 3, 5, 7, 6, 1)

To expand $t_n$ to become $t_{n+1}$, the random record $w(n + 1)$ has exactly $n + 1$ positions. Its relative rank will determine precisely which position will be selected.
Figure 16 m-ary trees with extremal path lengths

(a) $h_n = 0$
$H_n = 0$
$n = m - 1$

(b) $h_n = 1$
$H_n = \frac{n}{m - 1}$

(c) $h_n = \log_m(n + 1)$
$H_n = \log_m(n + 1)$
It is easily seen that

\[ 0 \leq h_n \leq \lceil \log_m(n + 1) \rceil, \]

\[ \lfloor \log_m(n + 1) \rfloor \leq H_n \leq \frac{n}{m - 1} \]

and all bounds are attainable (see figure 16); that is, there exist permutations \( w(n) \) so that \( t_n \) is one of the extremes. For example, if \( w(n) \) is sorted in ascending order, \( t_n \) will be that of figure 16(a).

The cost of searching for a record or inserting a new one depends on how deep we go into the tree, and of course, we wish to minimize this cost. Therefore, we would like our trees to be balanced and closer to complete trees than to a linear list. Thus the question of balancing naturally arises. We shall try in this chapter to adapt the fringe technique \([30],[31]\) to process m-ary trees.

### 5.2 Off-line balancing

We say that an m-ary tree is balanced if all paths from the root to a leaf node are of the same order of magnitude. A balancing algorithm which starts to work after seeing the entire input sequence is called an off-line algorithm. On the other hand, if balancing is done as the sequences progresses with time, the algorithm is said to be on-line.

In some applications, we may have the luxury of seeing the entire input before constructing the tree, and an off-line algorithm may be used. For example, when we construct the tree representing the file of records of the students of a school, we usually have a deadline to
receive applications, then at a later date, we process the subset of
the approved admissions.

A simple off-line balancing algorithm is given in figure 17.
The input is assumed to be stored in the array A of size n. A is
first sorted (this takes \( O(n \log n) \) using a standard sorting algorithm).
A is global, and we use GET NODE; the usual subroutine to seize an
available node from the pool of free nodes [9]. Each node is assumed
to include \( m - 1 \) INFO fields and \( m \) POINTER fields. The external call
to construct the balanced tree is

\[
\text{root} := \text{GETNODE};
\]
\[
\text{call SELECT_ROOT (1, n, root)};
\]

```
procedure SELECT_ROOT(L, U, root: integer);
1     if L > U then return;
2     if U - L + 1 <= m - 1 then
3       begin
4         for i := L to U - L + 1 do
5           NODE(root). INFO(i) := A[i];
6         return;
7       end;
8     else begin
9         for i := 1 to m - 1 do
10           NODE(root). INFO(i) := A[\( \frac{U - L + 1}{m} \) + L - 1];
11         for i := 0 to m - 2 do
12           begin
13             p := GETNODE;
14             NODE(root). POINTER(i + 1) := p;
15             call SELECT_ROOT(\[ \frac{U - L + 1}{m} \] + L, \[ \frac{(i + 1) \ U - L + 1}{m} \] + L - 2, p);
16           end;
17         end;
18         p := GETNODE;
19         NODE(root). POINTER(m) := p;
20         call SELECT_ROOT(\[ \frac{(m - 1) \ U - L + 1}{m} \] + L, U, p)
21       end.
```

Figure 17 An off-line balancing algorithm
Proof of correctness

We shall demonstrate that the algorithm of figure 17 always produces a complete tree.

If \( n \leq m - 1 \), then \( L = 1, U = n \), and the condition of step 1 is false, but that of step 2 is true \((U - L + 1 \leq m - 1)\). Then the elements \( A[1], \ldots, A[n] \) will label the root left to right and \( A[1] < A[2] < \cdots < A[n] \), since \( A \) is sorted. Then the algorithm returns without further ado; a tree on one node is a complete tree.

Assume \( n > m - 1 \); here we have two cases:

(a) \( n = m^g - 1 \), for some integer \( g \).

The algorithm will produce a perfect tree of height \( g - 1 \) in this case. (An \( m \)-ary tree is perfect if level \( i \) contains \( m^i(m - 1) \) keys, \( i = 0, 1, \ldots, g - 1 \), and \( n = m^g - 1 \).)

We shall prove this case by induction on \( g \). (\( g = 1 \) is the case \( n \leq m - 1 \) discussed above, and here the algorithm produces one node completely filled: a perfect tree of height zero.) Assume that for \( n = m^g - 1 \), \( g > 1 \), the algorithm produces a perfect tree of height \( g - 1 \). For \( n = m^{g+1} - 1 \), we have \( U = n, L = 1 \), and the conditions of steps 1 and 2 are obviously false, then steps 9 and 10 will select for the root the elements \( \Gamma_i \frac{U - L + 1}{m} + L - 1 = \Gamma_i \frac{m^{g+1} - 1}{m}, 1 \leq i \leq m - 2 \). The loop at step 11 then performs \( m - 1 \) calls to \text{GETNODE} \) and \( m - 1 \) recursive calls to \text{SELECT_ROOT} \), with the parameters

\[
L = \Gamma_i \frac{m^{g+1} - 1}{m} \quad \text{and} \quad U = \Gamma_i (i + 1) \frac{m^{g+1} - 1}{m}.
\]

But the \( i \)th call is a call to make a subtree from a subsection of the
array \ A of size

\[ \frac{(i + 1) m^{g+1} - 1}{m} - 2 - \left( \frac{m^{g+1} - 1}{m} + 1 \right) + 1 \]

\[ = (i + 1)m^g - m^g - 1 \]

\[ = m^g - 1. \]

Because \( A \) has been sorted, all the elements of the first subtree are
less than \( A[\frac{L - L + 1}{m} - 1] \), the rightmost key in the root of this
subtree. Similarly, all the elements of the \( j \)th subtree, \( 2 \leq j \leq m - 1 \),
lie properly between keys \( K_{j-1} \) and \( K_j \) of the root of the \( j \)th subtree.
Finally, steps 17 and 18 allocate a free node as the root of the \( m \)th
subtree, and the \( m \)th subtree is obtained by the recursive call at step
19; its size is also \( m^g - 1 \). By the induction hypothesis, each of the
trees \( T_{ni} \), \( 1 \leq i \leq m \), is a perfect tree of height \( g - 1 \); thus the tree
produced by the algorithm for \( n = m^{g+1} - 1 \) is a perfect tree of height \( g \).
This completes the induction.

(b) \( n \) is not of the form \( m^g - 1 \).

Simple reasoning shows that if the algorithm produces a complete
(but not perfect) tree on \( n \) \( (n \neq 1 \mod m^g) \), then the algorithm will
produce the same tree on \( n + 1 \) elements with \( w(n + 1) \) added to the
rightmost leaf node.

Q.E.D.

5.3 On-line balancing

In some applications, we may not be able to afford the luxury of
off-line balancing, and that is when the input records are to be processed as they progress in an ad hoc order over time. In this case, we may want to rebalance the tree with every new record if we do a lot of searching between insertions. For example, consider the file of identifications of the customers of a bank. In such a case, we may have a large number of queries to update accounts before a new customer opens an account, i.e., before making a new insertion. So the tree must be balanced or near balanced all the time.

A heuristic using an idea called the fringe technique was developed by Bell [30] and Walker and Wood [31] for binary trees, and later it was analyzed by Poblete and Munro [16]. The basic idea is to perform a local rotation at the bottom of the tree when we have a path subgraph of length two. See the example of figure 18.

![Diagram](image)

\[ x < y < z \]

(a) before rotation  \hspace{1cm} (b) after rotation

Figure 18  Local rotation at the fringe nodes to make the median key the root of the fringe subtree

This rotation prevents the development of long paths.
We shall employ this idea with m-way trees. If the tree gets "too deep" on one side, rebalance. The criterion for "too deep" is a matter of choice, and we take it to mean that a subgraph of the tree becomes a path of length \(m\) (\(m\) edges) with the last node completely filled. If an insertion creates such a path, then rotate the elements by first sorting them, then using steps 9 and 10 of the algorithm of figure 17.

select the elements of the root of this subtree, then all the elements between \(\frac{U - L + 1}{m} j\) and \(\frac{U - L + 1}{m} (j + 1)\), \(0 \leq j \leq m - 1\), will go to the \(j\)th subtree, counting subtrees from left to right.

**Example**

Assume a run of \(w\) is

5100, 100, 80, 4000, 300, 110, 150, 2000, 1400, 1000, 1300, 1200, 1010, 1080, 1020, 1150, 1090, 1120, 1110, 1100, 1140, 1135, 1130, 1125.

Let \(m = 5\); direct insertion without balancing will produce the tree of figure 19(a). The fringe heuristic will give the tree of figure 19(b).

This process is made easy if we add to each node two pointer fields:
- a count of the number of non-null pointers in the node (let us call it \text{SUBTREE\_COUNT});
- a backward pointer (\text{BACK\_POINTER}) to the father of the node.

Inserting a new element, we need a counter (\text{COUNTER}) for the length of the last path subgraph that will be traversed in walking down the tree to insert this element. \text{COUNTER} is initialized to zero, and whenever we advance one level in the tree, \text{SUBTREE\_COUNT} is checked. If it is exactly one, \text{COUNTER} is incremented by one, because this means one
Figure 19  The tree constructed from \( w \)
(a) using direct insertion
(b) using the fringe heuristic
more node in a path subgraph. If SUBTREE_COUNT is greater than one, COUNTER is reset to zero, as in figure 20.

![Counter changes diagram]

Figure 20  COUNTER changes according to the number of subtrees of each node

When the position for the new record is reached, the value of COUNTER is examined. If it is less than \(m\) or equal to \(m\), but the number of elements in the node it will belong to is less than \(m - 1\), no rotation will be needed. If its value is \(m\), and the node in which the new record is inserted will be completely filled, then a rotation is performed. Here we use BACK_POINTER to climb up \(m\) steps. The algorithm of figure 17 may then be used to rotate the fringe path subgraph into a perfect fringe subtree of height one.

Let us consider the worst case input and the amount of path
Figure 21  Direct insertion and the fringe heuristic in the worst case
compression produced by the fringe heuristic. If the input is sorted in ascending order, and \( n \) is not a multiple of \( m - 1 \), then naive insertion and the fringe heuristic will give the trees shown in figure 21(a) and 21(b), respectively.

Every \( m \) insertions (after the root is created), a rotation is needed. This will give \( \ell \) levels, each containing \( m \) nodes, and \( \alpha \) nodes, \( 0 \leq \alpha \leq m \), in a straight line. To find a statement relating \( \ell \) and \( \alpha \), notice first that the number of edges is the same in both trees. This number is \( H_n = \frac{n}{m - 1} \) for the tree of figure 20(a). The number of edges in the tree of figure 20(b) is \( \ell m + \alpha \) (each of the \( m \) nodes in the first \( \ell \) levels has one edge leading to it). Thus

\[
\ell m + \alpha = \frac{n}{m - 1},
\]

\( 0 \leq \alpha \leq m. \)

The height of the balanced tree is

\[
\bar{H}_n = \ell + \alpha
\]

\[
= \frac{1}{m} \left[ \ell m + \alpha + \alpha(m - 1) \right]
\]

\[
= \frac{1}{m} \left[ \ell m + \frac{n}{m - 1} + \frac{\alpha(m - 1)}{m} \right], \quad 0 \leq \alpha \leq m
\]

Thus

\[
\frac{\bar{H}_n}{H_n} = \frac{1}{m} + O\left(\frac{1}{n}\right)
\]

or

\[
\frac{\bar{H}_n}{H_n} \sim \frac{1}{m}.
\]

In practice, \( m \) is in the range of hundreds. For example, if \( m = 100 \), then the height of the balanced tree is 1% of the unbalanced tree. It
should also be noticed that if $n = \ell m(m - 1)$, a multiple of $m(m - 1)$, then the heuristic gives a tree of height $\ell$. Then if $n$ is gradually increased until it becomes the next multiple of $m(m - 1)$, then the height first increases until it becomes $\ell + m$ then drops to $\ell + 1$ when $n$ becomes $(\ell + 1)m(m - 1)$. Thus the fringe heuristic saves the height in the worst case.
In this chapter, we investigate the asymptotic behavior of some variables related to the performance of searching and insertion algorithms on m-ary trees constructed from random input sequences.

In an m-ary tree with n nodes $t_n$, let $L_n$ stand for the length of the path from the root to the node which will contain $w(n + 1)$ and let $C_n$ be the number of comparisons required to find the place for $w(n + 1)$. Of course, $C_n$ depends on how the current number of $w(n + 1)$ is compared with the elements in each node along the path to its destination. In practice, comparisons are made within each node using binary search in $O(\log m)$, and this is always possible since the elements are sorted in each node. (On the other hand, insertion takes $O(m)$ moves within each node.) In what follows, we assume that comparisons are made sequentially from left to right within each node as it becomes more tractable to analyze mathematically. It should be clear that the upper bound we get on $C_n$ under this assumption will also be an upper bound on $C_n$ if binary search is used, probably a very loose one.

We shall prove here, like binary trees constructed from random input sequences when all such sequences are equally likely, that the height and the shortest path from the root to a leaf node are both of logarithmic order. In fact, this will follow from our result on $L_n$.
that both *in probability* and *almost surely* $L_n$ is of logarithmic order.
Similar results are obtained for $C_n$.

### 6.1 Results and interpretations

In this chapter, we prove

**Theorem 6.1**

In probability

\[
\lim_{n \to \infty} \frac{L_n}{\ln n} = \frac{1}{\sum_{i=2}^{m} \frac{1}{i}} = a(m)
\]

\[
\lim_{n \to \infty} \frac{C_n}{\ln n} = \frac{(m-1)(m+2)}{2m} a(m)
\]

\[
= b(m).
\]

It will follow from the proof of theorem 6.1 that, with high probability, all but a negligible fraction of $n + 1$ paths leading to possible locations for $w(n + 1)$ have lengths between $(a(m) - \epsilon) \ln n$ and $(a(m) + \epsilon) \ln n$, $\forall \epsilon > 0$. Also, observe that $a(2) = b(2) = b(3) = 2 > a(3) = \frac{6}{5}$. Thus cramming two elements into each node instead of just one leads with high probability to noticeably shorter trees, while the typical values of $C_n$, the number of necessary comparisons, are kept essentially the same ($\sim 2 \ln n$).

While theorem 6.1 describes the property of almost all trees $t_n$ for a fixed (but large) $n$, theorems 6.2 and 6.3 are concerned with the
asymptotic behavior of almost all infinite input sequences. Theorems 6.2 and 6.3 tell us that for almost all infinite sequences, \( \frac{L_n}{\ln n} \) and \( \frac{C_n}{\ln n} \) are bounded below infinitely often, except for a finite number of values of \( n \), by constants, and bounded above infinitely often by two other constants. These constants are the roots of certain equations.

Theorem 6.2

With probability one,

\[
\begin{align*}
(6.3) \quad & \lim \inf \frac{L_n}{\ln n} \geq a_1, \\
(6.4) \quad & \lim \sup \frac{L_n}{\ln n} \leq a_2; \\
(6.5) \quad & a_1 = \frac{1}{m-2} \sum_{j=0}^{\infty} \frac{1}{x_i + j}
\end{align*}
\]

and \( 0 < x_1 < x_2 < \infty \) are the roots of the equation

\[
f(x) = x + \sum_{k=0}^{m-2} \ln \left( \frac{k + 2}{k + x} \right) - 1 = 0
\]

\[
\sum_{j=0}^{\infty} \frac{1}{x + j}
\]
Theorem 6.3

With probability one,

\[
\lim_{n \to \infty} \inf \frac{C_n}{\ln n} \geq \beta_1,
\]

(6.6)

\[
\lim_{n \to \infty} \sup \frac{C_n}{\ln n} \leq \beta_2,
\]

(6.7)

here \( \beta_1 = \beta(x_i) = y(x_i)/y'(x_i) \), \( y = y(x) \) being the positive root of the equation

\[
\sum_{j=1}^{m-1} y^j + y^{m-1} = x^{m-1},
\]

(6.8)

and \( 0 < x_1 < x_2 < \infty \) are the roots of the equation

\[
g(x) = x - \beta(x) \ln(y(x)) - 1 = 0.
\]

(6.9)

Loosely speaking, these statements mean that both \( L_n \) and \( C_n \) considered as random functions of \( n \) are almost surely bounded above and below by logarithmic functions of \( n \).

Notes (a) Consider the special case of the binary trees \( t_n \), i.e., \( m = 2 \). Then \( C_n = L_n \), and as should be expected, \( \beta_1 = \alpha_1, \beta_2 = \alpha_2 \), where (see (6.5), (6.8)) \( 0 < \alpha_1 < \alpha_2 < \infty \) are the roots of the equation

\[
\alpha + \alpha \ln(2/\alpha) - 1 = 0,
\]

\( \alpha_1 = 0.37, \alpha_2 = 4.31 \). The number \( \alpha_2 \) already appeared in [15], where \( H_n \), the height of \( t_n \), was shown not to exceed, with high probability, \( (\alpha_2 + \varepsilon) \ln n, \varepsilon > 0 \). There were also given in [15] some ingenious, though incomplete, arguments intended to prove that
E(H_n) \geq 3.63 \ln n + o(\ln n).

Pittel [22] has recently proved the existence of two constants $c_1 \in [0.37, 0.50]$ and $c_2 \in [3.58, 4.32]$ such that, with probability one,

$$\lim_{n \to \infty} \frac{h_n}{\ln n} = c_1, \quad \lim_{n \to \infty} \frac{H_n}{\ln n} = c_2.$$

As in chapter V, $h_n$ and $H_n$ are the lengths of the shortest and longest paths from the root of $t_n$ to the possible location of $w(n + 1)$. As $h_n \leq L_n \leq H_n$, and $L_n = h_n$ and $L_n = H_n$ infinitely often almost surely, we conclude that in this case, with probability one,

$$\liminf_{n \to \infty} \frac{L_n}{\ln n} = c_1, \quad \limsup_{n \to \infty} \frac{L_n}{\ln n} = c_2.$$

It is worth remembering (theorem 6.1) that still in probability

$$\lim_{n \to \infty} \frac{L_n}{\ln n} = 2 \in (c_1, c_2).$$

(b) Here is the table of $\alpha_i, \beta_i$, ($i = 1, 2$), rounded to three decimal places for some small values of $m$. 

<table>
<thead>
<tr>
<th>m</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.38</td>
<td>0.39</td>
<td>4.31</td>
<td>4.32</td>
</tr>
<tr>
<td>2</td>
<td>0.37</td>
<td>0.38</td>
<td>4.30</td>
<td>4.31</td>
</tr>
</tbody>
</table>
### TABLE 2

THE ROOTS OF (6.5) FOR SOME SMALL VALUES OF \( m \)

<table>
<thead>
<tr>
<th>( m )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.373</td>
<td>4.311</td>
<td>0.373</td>
<td>4.311</td>
</tr>
<tr>
<td>3</td>
<td>0.318</td>
<td>4.490</td>
<td>0.373</td>
<td>4.311</td>
</tr>
<tr>
<td>4</td>
<td>0.287</td>
<td>4.636</td>
<td>0.350</td>
<td>4.552</td>
</tr>
<tr>
<td>5</td>
<td>0.267</td>
<td>4.759</td>
<td>0.328</td>
<td>4.865</td>
</tr>
<tr>
<td>6</td>
<td>0.253</td>
<td>4.867</td>
<td>0.310</td>
<td>5.202</td>
</tr>
<tr>
<td>7</td>
<td>0.242</td>
<td>4.963</td>
<td>0.297</td>
<td>5.548</td>
</tr>
<tr>
<td>8</td>
<td>0.233</td>
<td>5.049</td>
<td>0.286</td>
<td>5.897</td>
</tr>
<tr>
<td>9</td>
<td>0.226</td>
<td>5.128</td>
<td>0.277</td>
<td>6.243</td>
</tr>
<tr>
<td>10</td>
<td>0.219</td>
<td>5.200</td>
<td>0.270</td>
<td>6.585</td>
</tr>
</tbody>
</table>

The table shows that \( a_1 = b_1 \) and \( a_2 = b_2 \) for \( m = 2 \); this must be expected as \( C_n = L_n \) in this case. What is surprising is the fact that \( b_1 \) and \( b_2 \) coincide for \( m = 2 \) and \( m = 3 \). This is a direct consequence of a stronger result which follows from Lemma 6.1 below: the distributions of \( C_n \) for \( m = 2 \) and \( m = 3 \) are the same.

### 6.2 Proofs

**Notations** Let \( X_{nk} \) be the random number of possible positions for \( w(n+1) \) at distance \( k \) from the root; clearly, \( X_{0k} = 6_{0k} \). Let \( Y_{nk} \) be the random number of possible positions for \( w(n+1) \) such that to reach
one of them $w(n + 1)$ has to be compared with exactly $k$ elements of $w(n) = (w(1), \ldots, w(n))$. Denote $F_{nk} = E(X_{nk})$, $G_{nk} = E(Y_{nk})$. The numbers $F_{nk}$, $G_{nk}$ are closely associated with the distributions of $L_n$ and $C_n$. Namely, according to the distribution of the vector of sequential ranks $r = (r(n))_{n=1}^\infty$,

$$P(L_n = k|t_n) = X_{nk}/(n + 1), P(C_n = k|t_n) = Y_{nk}/(n + 1);$$

so averaging over $t_n$ we have

(6.10)  \[ P(L_n = k) = F_{nk}/(n + 1), \quad P(C_n = k) = G_{nk}/(n + 1). \]

Introduce the generating functions of $F_{nk}$, $G_{nk}:

(6.11)  \begin{align*}
F_k(x) &= \sum_{n \geq 0} F_{nk} x^n, \\
F_n(y) &= \sum_{k \geq 0} F_{nk} y^k,
\end{align*}

$$F(x, y) = \sum_{n \geq 0} \sum_{k \geq 0} F_{nk} x^n y^k,$$

$$G_k(x) = \sum_{n \geq 0} G_{nk} x^n, \quad G_n(y) = \sum_{k \geq 0} G_{nk} y^k,$$

$$G(x, y) = \sum_{n \geq 0} \sum_{k \geq 0} G_{nk} x^n y^k.$$
with boundary conditions

\begin{equation}
\frac{a_{1}^{1}}{a_{x}^{1}} F(0,y) = (1 + 1)!, 0 < i < m-2,
\end{equation}

and

\begin{equation}
\frac{a_{m-1}^{m-1}}{a_{x}^{m-1}} G = (a_{m-1}^{y_1}(y) (m-1)!) G,
\end{equation}

with boundary conditions

\begin{equation}
\frac{a_{1}^{1}}{a_{x}^{1}} G(0,y) = i! \rho(y), 0 < i < m - 2,
\end{equation}

\begin{equation}
\rho_0(y) = 1, \rho_t(y) = y^t + \sum_{s=1}^{t} y^s \text{, for } t \geq 1.
\end{equation}

Proof (a) Observe first that

\begin{equation}
F_{n0} = n + 1 \text{ if } 0 < n < m - 2, \text{ and } F_{n0} = 0 \text{ if } n \geq m - 1,
\end{equation}

\begin{equation}
F_{nk} = 0 \text{ if } 0 < n < m - 2 \text{ and } k \geq 1.
\end{equation}

Let \( n \geq m - 1 \) and \( T_{n1}, \ldots, T_{nm} \) be the subtrees of \( t_n \) ordered from left to right, the roots of which are adjacent to the root of \( t_n \) (see introduction). If \( \tau_{nj} \) stands for the size of \( T_{nj} \), i.e., the number of elements of \( \omega(n) \) contained in its nodes, then

\begin{equation}
\sum_{j=1}^{m} \tau_{nj} = n - m + 1.
\end{equation}

Let us show that the random vector \( \tau_n = (\tau_{nj})_{j=1}^{m} \) is uniformly distributed on the set \( \Omega_{nm} \) of nonnegative integer-valued solutions of (6.19), in other words that
(6.20) \[ P(\tau_{nj} = i_j, 1 \leq j \leq m) = \left| \Omega_{nm} \right|^{-1} = \left( \begin{array}{c} n \\ m - 1 \end{array} \right)^{-1} \]

if

\[ \sum_{j=1}^{m} i_j = n - m + 1, \quad i_j \geq 0. \]

In order to get \( i_1, i_2, \ldots, i_m \) elements in subtrees \( T_{n1}, T_{n2}, \ldots, T_{nm} \), the elements with absolute ranks \( i_1 + 1, i_1 + i_2 + 2, \ldots \)

\[ i_1 + i_2 + \ldots + i_m + m - 1 \]

must appear in the root; see figure 22.

**Figure 22** An m-ary tree with subtrees of sizes \( i_1, i_2, \ldots, i_m \)
Any n-long permutation with these elements appearing as the initial segment (of length m-1) will give a tree with $i_1, i_2, ..., i_m$ elements in subtrees $T_{n1}, T_{n2}, ..., T_{nm}$. If this (m-1)-segment is permuted and the trailing segment of length n - m + 1 is permuted we still get a tree with $i_1, ..., i_m$ elements in subtrees $T_{n1}, T_{n2}, ..., T_{nm}$. But the initial (m-1) - long segment can be permuted in $(m-1)!$ ways and the trailing segment can be permuted in $(n - m + 1)!$ ways. Thus we have $(m - 1)! (n - m + 1)!$ ways out of n! that give the desired tree i.e.,

$$P(T_{ni} = i_j, 1 < j < m) = \frac{(m-1)! (n - m + 1)!}{n!}$$

$$= \left(\frac{n}{m-1}\right) .$$

Furthermore, for $k > 1$ (and $n > m - 1$)

$$x_{nk} = \sum_{j=1}^{m} x_{k-1}^{(j)} ,$$

where $x_{k-1}^{(j)}$ is the number of positions available for w(n + 1) in $T_{nj}$, which are $(k - 1)$ steps apart from its root. Notice that

$$P(x_{k-1} = a | T_{nj} = b) = P(x_{b,k-1} = a), 1 < j < m,$$

for all a and b. Now, taking the expectations of both sides of (6.21) we get
Hence

\[ F_{nk} = \sum_{j=1}^{m} E(X_{nk}) \]

Using (6.20), (6.22) and interchanging the order of summation

\[ F_{nk} = \sum_{i_1, i_2, \ldots, i_m} \sum_{j=1}^{m} \sum_{a=0}^{\infty} a P(X_{i_j, k-1} = a) \binom{n}{m-1} \]

or

\[ F_{nk} \left( \begin{array}{c} n \\ m-1 \end{array} \right) = \sum_{i_1, i_2, \ldots, i_m} \sum_{j=1}^{m} \sum_{a=0}^{\infty} a P(X_{i_j, k-1} = a) \]
Here the outer sum extends over all solutions of (6.19).

Therefore, using the notation \((n)_{m-1} = n(n - 1) \ldots (n - m + 2)\), we arrive at

\[
(6.24) \quad F_{k}^{(m-1)}(x) = \sum_{n \geq m-1} (n)_{m-1} F_{nk} x^{n-m+1} = (m-1)! \sum_{i_s \geq 0, 1 < s < m} \prod_{t=1}^{m} x^{i_t},
\]

or

\[
(6.25) \quad (1 - x)^{m-1} F_{k}^{(m-1)}(x) = m! F_{k-1}(x), \quad k \geq 1.
\]

Since, in view of (6.17),

\[
(6.26) \quad F_{0}(x) = \sum_{n \geq 0} F_{n0} x^{n} = \sum_{0 \leq n \leq m-2} (n + 1)x^{n},
\]

the relation (6.25) implies
\[(1 - x)^{m-1} e^{m-1} F(x,y)/e^{m-1} = (1 - x)^{m-1} \sum_{k \geq 1} F_k^{(m-1)}(x)y^k\]

= \((m!y) F(x,y)\).

Initial conditions (6.13) follow directly from (6.17), (6.18), and this completes the proof of part (a).

(b) Let \(Y_s(j), s \geq 0, 1 \leq j \leq m, \) be the number of positions available for \(w(n+1)\) in \(T_{nj}\) which are \(s\) comparisons away from the root of \(T_{nj}\) and let \(Y_s(j) = 0\) for \(s < 0\).

Now, for \(k > 1\) and \(n > m-1,\)

\[
Y_{nk} = \sum_{1 \leq j \leq m-1} Y_s(j) + Y_{k-m+1}.
\]

Since, to reach \(T_{nj}\) we have to consume \(j\) comparisons, then elements which are \(k-j\) comparisons away from the root of \(T_{nj}\) will be \(k\) comparisons away from the root of the whole tree. The term \(Y_{k-m+1}\) is for the right most tree \(T_{nm}\).

Define \(G_{nk} = 0\) for \(n > 0, k < 0\), then reasoning the same as in part (a) we reach

\[
G_k^{(m-1)}(x) = (m-1)! \cdot \sum_{\Pi_{t=1}^{m} i_t} \sum_{i_s > 0, 1 \leq s \leq m} G_{i_j, k-j} + G_{i_m, k-m+1}
\]
\[ (1-x)^{m-1} \sum_{j=1}^{m-1} G_{i,k-j} x^i + \sum_{i \geq 0} G_{i,k-m+1} x^i, \]

or

\[ (1-x)^{m-1} G_k^{(m-1)}(x) = (m-1)! \sum_{j=1}^{m-1} G_{k-j}(x) + G_{k-m+1}(x); \]

(in the right-hand expression, \( G_{s}(x) = 0 \) for \( s < 0 \)).

By definition of \( G_{nk} \), we also have \( G_{0k} = \delta_{0k} \), and, for

\( 1 \leq n \leq m-2, \)

\[ G_k = \begin{cases} 0, & \text{if } k = 0 \text{ or } k \geq n + 1, \\ 1, & \text{if } 1 \leq k \leq n - 1, \\ 2, & \text{if } k = n. \end{cases} \]

In particular, \( G_0(x) = 1 \), so according to (6.27),

\[ (1-x)^{m-1} x^{m-1} G(x,y)/x^{m-1} = (m-1)! \sum_{k \geq 1} y^k \]

\[ [ \sum_{j=1}^{m-1} G_{k-j}(x) + G_{k-m+1}(x) ] \]
\[ m - 1 \]
\[ = (m-1)! \left( \sum_{j=1}^{m-1} y^j \sum_{k > j} G_{k-j}(x)y^{k-j} + y^{m-1} \right) \]
\[ \sum_{k > m-1} G_{k-(m-1)}(x)y^{k-(m-1)} \]
\[ = (m-1)! \left( \sum_{j=1}^{m-1} y^j + y^{m-1} \right) G(x,y) = (m-1)! \rho_{m-1}(y) G(x,y). \]

Conditions (6.15) follow from (6.28). The lemma is thus proven.

**Lemma 6.2**  
Let \( y > 0 \) and \( \lambda = \lambda(y), \sigma = \sigma(y) \) be the positive roots of the equations

\[(6.29) \quad <z_{m-1} = y^{m-1}, <z_{m-1} = \rho_{m-1}(y)(m-1)! ; \]

Then

\[(6.30) \quad F_n(y) < \gamma <\lambda>_{n!/n!}, \gamma = \gamma(\lambda) = \max \left[ 1, (m-1)(m-2)/\lambda \right], \]
\[(6.31) \quad G_n(y) < \delta <\sigma>_{n!/n!}, \delta = \delta(m), \delta(2) = 1, \]

and (6.30), (6.31) become identities for \( m = 2 \).

**Remark**  
In the binary case (\( m = 2 \)), we have \( \lambda = \sigma = 2y \) and

\[ F_n(y) = G_n(y) = <2y>_{n!/n!}. \]

So, using a well known identity [24]
\[ \langle \xi \rangle_{\mu} = \sum_{\nu=0}^{\mu} S(\mu,\nu) \xi^\nu, \]

where \( S(\mu,\nu) \) are the Stirling numbers of the first kind, we get
(see (6.10), (6.11))

\[ F_{nk} = G_{nk} = 2^k S(n,k)/n! , \]

and

\[ P(L_n = k) = 2^k S(n,k)/(n+1)! , \]

which is Lynch's formula, [32].

Proof of lemma 6.2 As arguments for proving (6.30) and (6.31) are quite similar, we shall prove only (6.30). Notice first that

\[ F(x,y) = (1 - x)^{-\lambda} \]

is a solution of (6.12) which satisfies the conditions

(6.32) \[ \partial^i F(0,y)/\partial x^i = \lambda \cdot i, \ 0 < i < m - 2, \]

(\( \langle \lambda \rangle_0 = 1 \), by definition).

For the case \( m = 2 \)

\[ F(x,y) = F(x,y) \]

and subsequently,

\[ F_n(y) = \text{coeff}_x n (1 - x)^{-\lambda} = \langle \lambda \rangle_n /n! = \langle 2y \rangle_n /n! . \]
But for this case $y = \rho(\lambda) = \max [1,0] = 1$, hence, $F_n(y) = y^{\lambda \frac{n}{n!}}$,

and (6.31) is an identity for $m=2$.

Let now $m \geq 3$. Simple arguments based on (6.29) and comparison of (6.13) and (6.32) show that

(6.33) \[ \frac{\partial^i \tilde{F}(0,y)}{\partial x^i} \geq \frac{\partial^i F(0,y)}{\partial x^i}, \quad 0 \leq i \leq m - 2, \]

where

\[ \tilde{F}(x,y) = y^\gamma \frac{F(x,y)}{1-x} , \quad \gamma = \max [1, 2(m-1)(m-2)/y]. \]

Now, if

\[ \tilde{F}(x,y) = \sum_{n \geq 0} \tilde{F}_n(y)x^n, \]

then, (6.33) is equivalent to

(6.34) \[ F_n(y) \leq \tilde{F}_n(y) , \quad 0 \leq n \leq m - 2. \]

Next, we show that (6.34) is also true for $n \geq m - 1$. Both $\tilde{F}(x,y)$ and $F(x,y)$ as functions of $x$ are solutions of the equation

\[ \frac{d^{m-1}}{dx^{m-1}} \psi + \frac{(ym)! \psi}{(1-x)^{m-1}} = 0 \]

So, using Taylor's expansions of $F$ and $\tilde{F}$ about $x = 0$ and the binomial series
\[ \frac{1}{(1-x)^{m-1}} = \sum_{\mu=0}^{\infty} \binom{m+\mu-2}{m-2} x^\mu, \]

we obtain the recurrence relations: for \( n \geq m - 1 \),

\[ (n)_{m-1} \psi_n = (yml) \cdot \sum_{j=0}^{n-m+1} \binom{n-1-j}{m-2} \psi_j, \psi_n = \tilde{F}_n(y) \text{ or } F_n(y). \]

Invoking positivity of \((n)_{m-1}\) and \( \frac{n - 1 - j}{m - 2} \) for \( n \geq m - 1 \) and (6.34), by induction we conclude that

(6.35) \[ F_n(y) \leq \tilde{F}_n(y), \]

for all \( n \geq 0 \). The lemma is thus proved.

Proof of theorem 6.1

It follows from definitions (6.11) that

\[ F_{nk} y^k \leq F_n(y), \forall y > 0 \]

and by lemma (6.2)

\[ F_{nk} \leq \frac{\delta \ll \lambda \gg_n}{y^k n!} \]

or, equivalently,
observe that \( y = y(x) \) is strictly increasing,

(6.37) \[ y(0+) = 0, \ y(2) = 1, \ \lim_{x \to \infty} y(x) = \infty. \]

Introduce the function

(6.38) \[ \alpha = \alpha(x) = \frac{1}{x + \ldots + \frac{1}{x+m-2}}, \ x \in (0,\infty); \]

obviously \( \alpha(x) \) is also strictly increasing,

\[ \lim_{x \to 0^+} \alpha(x) = 0, \ \lim_{x \to \infty} \alpha(x) = \infty. \]

Denote also \( k(x,n) = \alpha(x) \ln n \). We want to show that

(6.39) \[ \sum_{k \geq k(x,n)} F_{nk} \leq \frac{\gamma_1(x) e^{f(x) \ln n}}{1 - \frac{1}{y}}, \text{ if } x > 2, \]

(6.40) \[ \sum_{k \geq k(x,n)} F_{nk} \leq \frac{\gamma_2(x) e^{f(x) \ln n}}{1 - y}, \text{ if } x < 2, \]

where \( f(x) = x - 1 - \alpha(x) \ln y \), and \( y = y(x) \) is defined in (6.36).

Consider the case \( x > 2 \). Clearly \( y = y(x) > 1 \), so by (6.36)
\[ \sum_{k > k(x,n)} F_{nk} \leq \frac{\gamma < x >_n}{n! (1 - \frac{1}{y}) y k(x,n)} \]

By the Stirling formula for the gamma-function

\[ \Gamma(z + 1) = \sqrt{2\pi z} \left( \frac{z}{e} \right)^z \left( 1 + o(1) \right), z \to \infty, \]

we have

\[ \frac{x^n}{n!} = \frac{x(x+1) \cdots (x+n-1)}{n!} \]

\[ = \frac{(x+n-1) \cdots x}{\Gamma(n+1)} \cdot \frac{(x-1)(x-2) \cdots (x-Lx)}{(x-1)(x-2) \cdots (x-Lx)} \]

\[ = \frac{\Gamma(x+n)}{\Gamma(x) \Gamma(n+1)} \]

\[ = \frac{1}{\Gamma(x)} \left( \frac{x+n-1}{e} \right)^{x+n-1} / (\frac{n}{e})^n (1 + o(1)), \text{ as } n \to \infty \]

\[ = \frac{1}{\Gamma(x)} \left( 1 + \frac{x-1}{n} \right)^n (\frac{x+n-1}{e})^{x-1} (1 + o(1)) \]

\[ = \frac{1}{\Gamma(x)} e^{x-1} \frac{n^{x-1}}{e^{x-1}} (1 + o(1)) \]

\[ = \frac{1}{\Gamma(x)} e^{(x-1) \ln n} (1 + o(1)). \]

Therefore

\[ \sum_{k > k(x,n)} F_{nk} \leq \frac{1}{y_k} \gamma_1 e^{(x-1) \ln n}, \gamma_1(x) \text{ is a function of } x \text{ only.} \]
\begin{equation}
\leq \frac{\gamma_1(x)}{(1 - \frac{1}{y})} e^{(x-1)\ln n - k \ln y}
\end{equation}

\begin{equation}
= \frac{\gamma_1(x)}{(1 - \frac{1}{y})} e^{f(x)\ln n}
\end{equation}

The case $x < 2$ is treated similarly.

It is easy to check that

\begin{equation}
limit_{x \to 0^+} f(x) = -1, f(2) = 1, \lim_{x \to \infty} f(x) = -\infty.
\end{equation}

and

\begin{equation}
f'(x) = \frac{-\frac{2}{x^2} + \cdots + \frac{1}{(x+m-2)^2}}{x^2 + \cdots + (x+m-2)^2}
\end{equation}

so $f(x)$ is unimodal and

\begin{equation}
\max \{f(x) : 0 < x < \infty\} = f(2) = 1.
\end{equation}

Then in view of (6.10) and (6.39) - (4.43), for all $0 < x' < 2 < x'' < \infty$,

\begin{equation}
P(L_n \leq a(x')\ln n, \text{ or } L_n > a(x'')\ln n)
\end{equation}

\begin{equation}
\leq n^{-1}(\sum_{k < k(x',n)} F_{nk} + \sum_{k > k(x'',n)} F_{nk})
\end{equation}

\begin{equation}
\leq \gamma(x',x'') \exp[(\ln n) \max (f(x') - 1, f(x'') - 1)] \to 0,
\end{equation}

as $n \to \infty$. Since
\[ a(2) = \frac{1}{2 + \cdots + \frac{1}{m}} \]

It follows then that, in probability

\[ \frac{L_n}{\ln n} \to a(2). \]

(b) Similar to (6.36)

\[ G_{nk} < \langle x \rangle_n^{1/y} y^n, \quad x > 0, \]

where \( y \) is the positive root of

\[ p_{m-1}(y)(m-1)! = \langle x \rangle_{m-1}, \quad p_{m-1}(y) = \sum_{j=1}^{m-1} y^j + y^{m-1}, \]

which shows that \( y = y(x) \) is strictly increasing and satisfies (6.37).

It can be also verified that \( y'(x) \in [C_1, C_2] \) where \( C_1 < C_2 \) are two positive constants. Introduce the function

\[ \beta(x) = \frac{y(x)}{y'(x)}; \]

clearly, \( \beta(0+) = 0 \), \( \lim_{x \to \infty} \beta(x) = \infty \). Then, exactly as in (a), one can show that

\[ \sum_{k \geq x(x,n)} G_{nk} < \frac{a_1(x) \exp[g(x)\ln n]}{1 - \frac{1}{y}}, \quad \text{if } x > 2, \]
(6.50) \[ \sum_{k \leq \chi(x,n)} g_{nk} < \frac{\delta_2(x) \exp[g(x)\ln n]}{1 - y}, \text{ if } x < 2, \]

(6.51) \[ x(x,n) = \beta(x)\ln n, \ g(x) = x - 1 - \beta(x)\ln y. \]

Now, as in (6.41),
\[ g(0^+) = -1, \ g(2) = 1, \ \lim_{x \to \infty} g(x) = -\infty. \]

Let us demonstrate also that, like \( f(x) \), \( g(x) \) is unimodal. We have (see (6.51)),
\[ g'(x) = 1 - \beta'(x)\ln y(x) = \beta(x)y'(x)/y(x) = -\beta'(x)\ln y(x). \]

Since \( y(x) = 1 \) iff \( x = 2 \), unimodality will follow if we prove that \( \beta'(x) > 0 \) for \( x \in (0, \infty) \). Taking the logarithmic derivative of both sides of (6.47) leads to (see (6.48)),

(6.52) \[ \beta(x) = \frac{y \rho'_{m-1}(y)}{\rho_{m-1}(y) \left[ \frac{1}{x} + \cdots + \frac{1}{x+m-2} \right]} . \]

Notice that \( y'(x) > 0 \), and
\[ \frac{1}{x} + \cdots + \frac{1}{x+m-2} \]

has a positive derivative, too.
Now as \( \rho_{m-1}(y) = \sum_{j=1}^{m-1} \omega_j y^j \), \( \omega_j > 0 \), we also have

\[
d(y \rho'_{m-1}(y) \rho_{m-1}^{-1}(y))/dy =
\]

\[
[y \rho_{m-1}^2(y)]^{-1} \left( \sum_{j=1}^{m} j^2 \omega_j y^j \right) \left( \sum_{j=1}^{m} \omega_j y^j \right) - (\sum_{j=1}^{m} \omega_j y^j)^2 > 0,
\]

by the Cauchy-Schwarz inequality. Thus \( \beta'(x) > 0 \) is unimodal and

\[
\max \{g(x) : 0 < x < \infty\} = g(2) = 1.
\]

The rest of the proof goes exactly as in (a), and we get: in probability,

\[
\frac{C_n}{\ln n} \to \beta(2),
\]

where (as \( y(2) = 1 \))

\[
\beta(2) = \frac{\rho'_{m-1}(1)}{\rho_{m-1}(1) \left[ \frac{1}{2} + \cdots + \frac{1}{m} \right]}
\]

\[
= \frac{(m-1)(m+2)}{2m} \frac{1}{\frac{2}{2} + \cdots + \frac{1}{m}}.
\]

Theorem 6.1 is thus proven.
(6.53) \( h_n \leq L_n \leq H_n \).

(a) According to the proof of Theorem 1, \( f(x) \) is unimodal, and there exist two positive roots \( 0 < x_1 < 2 < x_2 \) of \( f(x) = 0 \), so that \( f(x) > 0 \) for \( x \in (x_1, x_2) \) and \( f(x) < 0 \) for \( x \in (x_1, x_2)^C \).

Given \( c > 0 \), introduce \( k(c,n) = a(x_2 + c) \ln n \), \( a(x) \) being defined in (6.38). As \( x_2 > 2 \), by (6.39) we have

\[
P(H_n > K(c,n)) = P\left( \bigcup_{k > k(c,n)} (X_{nk} > 0) \right)
\]

\[
< \sum_{k > k(c,n)} P(X_{nk} > 0) < \sum_{k > k(c,n)} E(X_{nk}) = \sum_{k > (c,n)} F_{nk}
\]

\[
< c \exp[f(x_2 + c) \ln n] = cn^{-c_1} , \quad c = c(c) > 0 , \quad c_1 = c_1(c) > 0 .
\]

So

(6.54) \( P(H_n > a(x_2 + c) \ln n) < cn^{-c_1} . \)

Similarly,

(6.55) \( P(h_n < a(x_1 - c) \ln n) < dn^{-d_1} , \quad d = d(c) > 0 , \quad d_1 = d_1(c) > 0 . \)

(b) Let us show that (6.54) implies that

\[
P(\limsup_{n \to \infty} \frac{H_n}{\ln n} < a_2) = 1 ,
\]

in other words that, for each \( a > a_2 \),
\[ P(H_n \geq \alpha \ln n \text{ infinitely often}) = 0. \]

To this end it would suffice to show (Borel-Cantelli lemma [26]) that

\[ \sum_{n=1}^{\infty} P(H_n \geq \alpha \ln n) < \infty. \]

As in the proof of theorem 4.2 it follows that with probability one

\[ \lim \inf_n \frac{L_n}{\ln n} \geq \alpha_1, \lim \sup_n \frac{L_n}{\ln n} \leq \alpha_2. \]

Theorem 6.2 is thus proven.

The proof of theorem 6.3 can be done essentially in the same way and we omit it.
CHAPTER VII
CONCLUSION AND TOPICS
FOR FUTURE RESEARCH

It has been demonstrated in this work that the risk involved is minimal when we construct m-ary search trees from random input sequences without balancing the resulting trees, because such trees are well balanced for almost all infinite and sufficiently long finite sequences. (All paths from the root to a leaf node are $O(\log n)$.) The methods used and the results obtained lead us to ask a number of questions that are interesting from the viewpoints of both computer science and mathematics.

It was shown in chapter II that for every binary tree, we have at least one permutation realizing it. We also know that more than one permutation may give rise to the same tree.

But what characterizes the permutations that give the same tree?

Also in chapter II, several random variables concerning the properties of binary search trees were defined. Their expected values were found.

What can we say about higher moments and distributions of these random variables?

In chapter V, we tried to adapt the fringe heuristic to m-ary trees. The heuristic which was suggested waits until we have $m + 1$
nodes forming a path subgraph.

Can we do any better if we balance the tree sooner than this?

Of course, a rigorous mathematical analysis of the expected height and its variance will be the key to the answer.

The number of comparisons needed to insert a random record was analyzed in chapter VI. The assumption was that comparisons are made sequentially within each node, because it was a more tractable mathematical problem. The bound obtained is also a bound if binary search is applied, which is a common practice.

Is this too loose a bound?

In chapter IV, we found the generating functions for the sequences of expected values of some random variables defined on m-ary trees. These generating functions were all partial differential equations of order m - 1. The asymptotic analysis of solutions of these equations led us to draw conclusions about the time required for searching and insertion.

Using techniques similar to those of chapter VI, we can show that the generating function of sequences of the expected random internal and external path lengths as well as the expected value of the random number of nodes are also partial differential equations of order (m - 1). The first two random variables are measures of the average cost of successful and unsuccessful searches. The latter is a measure of the average space required to store m-ary trees. It remains to analyze the asymptotic behavior of these random variables by analyzing solutions of the generating functions.

Finally, we hope that the methods and the results of this work
can be extended to other classes of random trees to answer questions similar to those we have answered for m-ary search trees constructed from random permutations.
APPENDIX A

COMPUTER RESULTS CONCERNING
THE HEIGHT OF A BST

TABLE 3
THE AVERAGE AND VARIANCE OF THE HEIGHT
OF A BST ON \( n \) NODES AS COMPARED TO \( \ln n \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( d_n )</th>
<th>( \ln n )</th>
<th>( \frac{d_n}{\ln n} )</th>
<th>( \text{var}(d_n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>( \approx )</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>2.000</td>
<td>0.693</td>
<td>2.885</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>2.667</td>
<td>1.099</td>
<td>2.427</td>
<td>0.222</td>
</tr>
<tr>
<td>4</td>
<td>3.333</td>
<td>1.386</td>
<td>2.404</td>
<td>0.222</td>
</tr>
<tr>
<td>5</td>
<td>3.800</td>
<td>1.609</td>
<td>2.361</td>
<td>0.427</td>
</tr>
<tr>
<td>6</td>
<td>4.267</td>
<td>1.792</td>
<td>2.381</td>
<td>0.507</td>
</tr>
<tr>
<td>7</td>
<td>4.686</td>
<td>1.946</td>
<td>2.408</td>
<td>0.533</td>
</tr>
<tr>
<td>8</td>
<td>5.021</td>
<td>2.079</td>
<td>2.415</td>
<td>0.678</td>
</tr>
<tr>
<td>9</td>
<td>5.394</td>
<td>2.197</td>
<td>2.455</td>
<td>0.682</td>
</tr>
<tr>
<td>10</td>
<td>5.656</td>
<td>2.303</td>
<td>2.456</td>
<td>0.827</td>
</tr>
<tr>
<td>11</td>
<td>6.011</td>
<td>2.398</td>
<td>2.507</td>
<td>0.789</td>
</tr>
<tr>
<td>12</td>
<td>6.200</td>
<td>2.485</td>
<td>2.495</td>
<td>0.952</td>
</tr>
<tr>
<td>13</td>
<td>6.543</td>
<td>2.565</td>
<td>2.551</td>
<td>0.943</td>
</tr>
<tr>
<td>( n )</td>
<td>( d_n )</td>
<td>( \ln n )</td>
<td>( \frac{d_n}{\ln n} )</td>
<td>( \text{var}(d_n) )</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>14</td>
<td>6.677</td>
<td>2.639</td>
<td>2.530</td>
<td>1.081</td>
</tr>
<tr>
<td>15</td>
<td>7.019</td>
<td>2.708</td>
<td>2.592</td>
<td>1.111</td>
</tr>
<tr>
<td>16</td>
<td>7.107</td>
<td>2.773</td>
<td>2.563</td>
<td>1.195</td>
</tr>
<tr>
<td>17</td>
<td>7.447</td>
<td>2.833</td>
<td>2.628</td>
<td>1.306</td>
</tr>
<tr>
<td>18</td>
<td>7.494</td>
<td>2.890</td>
<td>2.593</td>
<td>1.312</td>
</tr>
<tr>
<td>19</td>
<td>7.835</td>
<td>2.944</td>
<td>2.661</td>
<td>1.538</td>
</tr>
<tr>
<td>20</td>
<td>7.849</td>
<td>2.996</td>
<td>2.620</td>
<td>1.437</td>
</tr>
<tr>
<td>21</td>
<td>8.192</td>
<td>3.045</td>
<td>2.691</td>
<td>1.793</td>
</tr>
<tr>
<td>22</td>
<td>8.177</td>
<td>3.091</td>
<td>2.645</td>
<td>1.564</td>
</tr>
<tr>
<td>23</td>
<td>8.521</td>
<td>3.135</td>
<td>2.717</td>
<td>2.082</td>
</tr>
<tr>
<td>24</td>
<td>8.482</td>
<td>3.178</td>
<td>2.669</td>
<td>1.696</td>
</tr>
<tr>
<td>25</td>
<td>8.829</td>
<td>3.219</td>
<td>2.743</td>
<td>2.384</td>
</tr>
<tr>
<td>26</td>
<td>8.766</td>
<td>3.258</td>
<td>2.691</td>
<td>1.837</td>
</tr>
<tr>
<td>27</td>
<td>9.114</td>
<td>3.296</td>
<td>2.765</td>
<td>2.730</td>
</tr>
<tr>
<td>28</td>
<td>9.033</td>
<td>3.332</td>
<td>2.711</td>
<td>1.987</td>
</tr>
<tr>
<td>29</td>
<td>9.386</td>
<td>3.367</td>
<td>2.788</td>
<td>3.079</td>
</tr>
<tr>
<td>30</td>
<td>9.285</td>
<td>3.401</td>
<td>2.730</td>
<td>2.145</td>
</tr>
<tr>
<td>31</td>
<td>9.637</td>
<td>3.434</td>
<td>2.806</td>
<td>3.473</td>
</tr>
<tr>
<td>32</td>
<td>9.524</td>
<td>3.466</td>
<td>2.748</td>
<td>2.311</td>
</tr>
<tr>
<td>33</td>
<td>9.883</td>
<td>3.497</td>
<td>2.826</td>
<td>3.864</td>
</tr>
<tr>
<td>34</td>
<td>9.751</td>
<td>3.526</td>
<td>2.765</td>
<td>2.487</td>
</tr>
<tr>
<td>35</td>
<td>10.108</td>
<td>3.555</td>
<td>2.843</td>
<td>4.300</td>
</tr>
<tr>
<td>36</td>
<td>9.967</td>
<td>3.584</td>
<td>2.781</td>
<td>2.671</td>
</tr>
<tr>
<td>37</td>
<td>10.331</td>
<td>3.611</td>
<td>2.861</td>
<td>4.732</td>
</tr>
<tr>
<td>$n$</td>
<td>$d_n$</td>
<td>$\ln n$</td>
<td>$\frac{d_n}{\ln n}$</td>
<td>$\text{var}(d_n)$</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-----------</td>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>38</td>
<td>10.173</td>
<td>3.638</td>
<td>2.797</td>
<td>2.864</td>
</tr>
<tr>
<td>39</td>
<td>10.535</td>
<td>3.664</td>
<td>2.876</td>
<td>5.205</td>
</tr>
<tr>
<td>40</td>
<td>10.371</td>
<td>3.689</td>
<td>2.811</td>
<td>3.067</td>
</tr>
<tr>
<td>41</td>
<td>10.740</td>
<td>3.714</td>
<td>2.892</td>
<td>5.677</td>
</tr>
<tr>
<td>42</td>
<td>10.560</td>
<td>3.738</td>
<td>2.825</td>
<td>3.279</td>
</tr>
<tr>
<td>43</td>
<td>10.926</td>
<td>3.761</td>
<td>2.905</td>
<td>6.182</td>
</tr>
<tr>
<td>44</td>
<td>10.743</td>
<td>3.784</td>
<td>2.839</td>
<td>3.500</td>
</tr>
<tr>
<td>45</td>
<td>11.117</td>
<td>3.807</td>
<td>2.920</td>
<td>6.691</td>
</tr>
<tr>
<td>46</td>
<td>10.919</td>
<td>3.829</td>
<td>2.852</td>
<td>3.730</td>
</tr>
<tr>
<td>47</td>
<td>11.289</td>
<td>3.850</td>
<td>2.932</td>
<td>7.225</td>
</tr>
<tr>
<td>48</td>
<td>11.088</td>
<td>3.871</td>
<td>2.864</td>
<td>3.969</td>
</tr>
<tr>
<td>56</td>
<td>11.714</td>
<td>4.025</td>
<td>2.910</td>
<td>5.011</td>
</tr>
<tr>
<td>64</td>
<td>12.270</td>
<td>4.159</td>
<td>2.950</td>
<td>6.186</td>
</tr>
<tr>
<td>72</td>
<td>12.773</td>
<td>4.277</td>
<td>2.987</td>
<td>7.486</td>
</tr>
<tr>
<td>80</td>
<td>13.233</td>
<td>4.382</td>
<td>3.020</td>
<td>8.903</td>
</tr>
<tr>
<td>88</td>
<td>13.658</td>
<td>4.477</td>
<td>3.051</td>
<td>10.431</td>
</tr>
<tr>
<td>96</td>
<td>14.053</td>
<td>4.564</td>
<td>3.079</td>
<td>12.062</td>
</tr>
<tr>
<td>112</td>
<td>14.772</td>
<td>4.718</td>
<td>3.131</td>
<td>15.611</td>
</tr>
<tr>
<td>120</td>
<td>15.102</td>
<td>4.787</td>
<td>3.154</td>
<td>17.518</td>
</tr>
<tr>
<td>128</td>
<td>15.415</td>
<td>4.852</td>
<td>3.177</td>
<td>19.502</td>
</tr>
<tr>
<td>136</td>
<td>15.714</td>
<td>4.913</td>
<td>3.199</td>
<td>21.569</td>
</tr>
<tr>
<td>144</td>
<td>16.000</td>
<td>4.970</td>
<td>3.219</td>
<td>23.711</td>
</tr>
<tr>
<td>152</td>
<td>16.274</td>
<td>5.024</td>
<td>3.239</td>
<td>25.943</td>
</tr>
<tr>
<td>n</td>
<td>$d_n$</td>
<td>$\ln n$</td>
<td>$\frac{d_n}{\ln n}$</td>
<td>var($d_n$)</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>---------</td>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>160</td>
<td>16.537</td>
<td>5.075</td>
<td>3.258</td>
<td>28.226</td>
</tr>
<tr>
<td>168</td>
<td>16.791</td>
<td>5.124</td>
<td>3.277</td>
<td>30.577</td>
</tr>
<tr>
<td>176</td>
<td>17.036</td>
<td>5.170</td>
<td>3.295</td>
<td>32.993</td>
</tr>
<tr>
<td>184</td>
<td>17.273</td>
<td>5.215</td>
<td>3.312</td>
<td>35.467</td>
</tr>
<tr>
<td>192</td>
<td>17.503</td>
<td>5.257</td>
<td>3.329</td>
<td>37.997</td>
</tr>
<tr>
<td>200</td>
<td>17.726</td>
<td>5.298</td>
<td>3.346</td>
<td>40.580</td>
</tr>
<tr>
<td>225</td>
<td>18.807</td>
<td>5.416</td>
<td>3.472</td>
<td>88.305</td>
</tr>
<tr>
<td>250</td>
<td>18.992</td>
<td>5.521</td>
<td>3.440</td>
<td>57.895</td>
</tr>
<tr>
<td>275</td>
<td>19.990</td>
<td>5.617</td>
<td>3.559</td>
<td>17.591</td>
</tr>
<tr>
<td>300</td>
<td>20.094</td>
<td>5.704</td>
<td>3.523</td>
<td>76.951</td>
</tr>
<tr>
<td>325</td>
<td>21.038</td>
<td>5.784</td>
<td>3.637</td>
<td>48.877</td>
</tr>
<tr>
<td>350</td>
<td>21.078</td>
<td>5.858</td>
<td>3.598</td>
<td>97.450</td>
</tr>
<tr>
<td>375</td>
<td>21.978</td>
<td>5.927</td>
<td>3.708</td>
<td>81.343</td>
</tr>
<tr>
<td>400</td>
<td>21.974</td>
<td>5.991</td>
<td>3.667</td>
<td>19.190</td>
</tr>
<tr>
<td>425</td>
<td>22.844</td>
<td>6.052</td>
<td>3.775</td>
<td>15.373</td>
</tr>
<tr>
<td>450</td>
<td>22.801</td>
<td>6.109</td>
<td>3.732</td>
<td>42.047</td>
</tr>
<tr>
<td>475</td>
<td>23.644</td>
<td>6.163</td>
<td>3.836</td>
<td>50.202</td>
</tr>
<tr>
<td>500</td>
<td>23.572</td>
<td>6.215</td>
<td>3.793</td>
<td>65.884</td>
</tr>
<tr>
<td>525</td>
<td>24.397</td>
<td>6.263</td>
<td>3.895</td>
<td>86.623</td>
</tr>
<tr>
<td>550</td>
<td>24.301</td>
<td>6.310</td>
<td>3.851</td>
<td>91.606</td>
</tr>
<tr>
<td>600</td>
<td>24.994</td>
<td>6.397</td>
<td>3.907</td>
<td>19.099</td>
</tr>
<tr>
<td>625</td>
<td>25.789</td>
<td>6.438</td>
<td>4.006</td>
<td>65.156</td>
</tr>
<tr>
<td>650</td>
<td>25.655</td>
<td>6.477</td>
<td>3.961</td>
<td>48.182</td>
</tr>
<tr>
<td>n</td>
<td>$d_n$</td>
<td>$\ln n$</td>
<td>$\frac{d_n}{\ln n}$</td>
<td>$\text{var}(d_n)$</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>---------</td>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>675</td>
<td>26.436</td>
<td>6.515</td>
<td>4.058</td>
<td>06.652</td>
</tr>
<tr>
<td>700</td>
<td>26.288</td>
<td>6.551</td>
<td>4.013</td>
<td>78.856</td>
</tr>
<tr>
<td>725</td>
<td>27.060</td>
<td>6.586</td>
<td>4.109</td>
<td>50.156</td>
</tr>
<tr>
<td>775</td>
<td>27.659</td>
<td>6.653</td>
<td>4.157</td>
<td>94.903</td>
</tr>
<tr>
<td>800</td>
<td>27.486</td>
<td>6.685</td>
<td>4.112</td>
<td>45.015</td>
</tr>
<tr>
<td>825</td>
<td>28.240</td>
<td>6.715</td>
<td>4.205</td>
<td>41.682</td>
</tr>
<tr>
<td>850</td>
<td>28.055</td>
<td>6.745</td>
<td>4.159</td>
<td>80.499</td>
</tr>
<tr>
<td>875</td>
<td>28.800</td>
<td>6.774</td>
<td>4.251</td>
<td>89.658</td>
</tr>
<tr>
<td>900</td>
<td>28.607</td>
<td>6.802</td>
<td>4.205</td>
<td>17.520</td>
</tr>
<tr>
<td>925</td>
<td>29.347</td>
<td>6.830</td>
<td>4.297</td>
<td>39.531</td>
</tr>
<tr>
<td>950</td>
<td>29.143</td>
<td>6.856</td>
<td>4.250</td>
<td>56.024</td>
</tr>
<tr>
<td>975</td>
<td>29.876</td>
<td>6.882</td>
<td>4.341</td>
<td>90.574</td>
</tr>
<tr>
<td>1000</td>
<td>29.665</td>
<td>6.908</td>
<td>4.294</td>
<td>96.081</td>
</tr>
</tbody>
</table>
**APPENDIX B**

**COMPUTER RESULTS CONCERNING THE SHORTEST PATH**

**FROM THE ROOT TO A LEAF NODE IN A BST**

**TABLE 4**

*THE AVERAGE AND VARIANCE OF THE SHORTEST PATH FROM THE ROOT TO A LEAF NODE IN A BST ON n NODES AS COMPARED TO ln n*

<table>
<thead>
<tr>
<th>n</th>
<th>$S_n$</th>
<th>$\ln n$</th>
<th>$\frac{S_n}{\ln n}$</th>
<th>var($S_n$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>$\infty$</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000</td>
<td>0.693</td>
<td>1.44</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>1.333</td>
<td>1.099</td>
<td>1.21</td>
<td>0.222</td>
</tr>
<tr>
<td>4</td>
<td>1.500</td>
<td>1.386</td>
<td>1.08</td>
<td>0.250</td>
</tr>
<tr>
<td>5</td>
<td>1.600</td>
<td>1.609</td>
<td>0.99</td>
<td>0.240</td>
</tr>
<tr>
<td>6</td>
<td>1.667</td>
<td>1.792</td>
<td>0.93</td>
<td>0.222</td>
</tr>
<tr>
<td>7</td>
<td>1.714</td>
<td>1.946</td>
<td>0.88</td>
<td>0.204</td>
</tr>
<tr>
<td>8</td>
<td>1.792</td>
<td>2.079</td>
<td>0.86</td>
<td>0.248</td>
</tr>
<tr>
<td>9</td>
<td>1.822</td>
<td>2.197</td>
<td>0.83</td>
<td>0.235</td>
</tr>
<tr>
<td>10</td>
<td>1.904</td>
<td>2.303</td>
<td>0.83</td>
<td>0.295</td>
</tr>
<tr>
<td>11</td>
<td>1.922</td>
<td>2.398</td>
<td>0.80</td>
<td>0.280</td>
</tr>
<tr>
<td>12</td>
<td>2.001</td>
<td>2.485</td>
<td>0.81</td>
<td>0.335</td>
</tr>
<tr>
<td>13</td>
<td>2.010</td>
<td>2.565</td>
<td>0.78</td>
<td>0.317</td>
</tr>
<tr>
<td>n</td>
<td>$S_n$</td>
<td>$ln\ n$</td>
<td>$\frac{S_n}{ln\ n}$</td>
<td>var($S_n$)</td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
<td>---------</td>
<td>----------------------</td>
<td>------------</td>
</tr>
<tr>
<td>14</td>
<td>2.083</td>
<td>2.639</td>
<td>0.79</td>
<td>0.362</td>
</tr>
<tr>
<td>15</td>
<td>2.085</td>
<td>2.708</td>
<td>0.77</td>
<td>0.345</td>
</tr>
<tr>
<td>16</td>
<td>2.153</td>
<td>2.773</td>
<td>0.78</td>
<td>0.379</td>
</tr>
<tr>
<td>17</td>
<td>2.150</td>
<td>2.833</td>
<td>0.76</td>
<td>0.363</td>
</tr>
<tr>
<td>18</td>
<td>2.212</td>
<td>2.890</td>
<td>0.77</td>
<td>0.390</td>
</tr>
<tr>
<td>19</td>
<td>2.207</td>
<td>2.944</td>
<td>0.75</td>
<td>0.376</td>
</tr>
<tr>
<td>20</td>
<td>2.264</td>
<td>2.996</td>
<td>0.76</td>
<td>0.396</td>
</tr>
<tr>
<td>21</td>
<td>2.257</td>
<td>3.045</td>
<td>0.74</td>
<td>0.384</td>
</tr>
<tr>
<td>22</td>
<td>2.310</td>
<td>3.091</td>
<td>0.75</td>
<td>0.400</td>
</tr>
<tr>
<td>23</td>
<td>2.302</td>
<td>3.135</td>
<td>0.73</td>
<td>0.391</td>
</tr>
<tr>
<td>24</td>
<td>2.352</td>
<td>3.178</td>
<td>0.74</td>
<td>0.404</td>
</tr>
<tr>
<td>25</td>
<td>2.343</td>
<td>3.219</td>
<td>0.73</td>
<td>0.395</td>
</tr>
<tr>
<td>26</td>
<td>2.390</td>
<td>3.258</td>
<td>0.73</td>
<td>0.407</td>
</tr>
<tr>
<td>27</td>
<td>2.380</td>
<td>3.296</td>
<td>0.72</td>
<td>0.401</td>
</tr>
<tr>
<td>28</td>
<td>2.425</td>
<td>3.332</td>
<td>0.73</td>
<td>0.410</td>
</tr>
<tr>
<td>29</td>
<td>2.414</td>
<td>3.367</td>
<td>0.72</td>
<td>0.404</td>
</tr>
<tr>
<td>30</td>
<td>2.457</td>
<td>3.401</td>
<td>0.72</td>
<td>0.413</td>
</tr>
<tr>
<td>31</td>
<td>2.447</td>
<td>3.434</td>
<td>0.71</td>
<td>0.410</td>
</tr>
<tr>
<td>32</td>
<td>2.487</td>
<td>3.466</td>
<td>0.72</td>
<td>0.417</td>
</tr>
<tr>
<td>33</td>
<td>2.476</td>
<td>3.497</td>
<td>0.71</td>
<td>0.414</td>
</tr>
<tr>
<td>34</td>
<td>2.516</td>
<td>3.526</td>
<td>0.71</td>
<td>0.421</td>
</tr>
<tr>
<td>35</td>
<td>2.505</td>
<td>3.555</td>
<td>0.70</td>
<td>0.420</td>
</tr>
<tr>
<td>36</td>
<td>2.542</td>
<td>3.584</td>
<td>0.71</td>
<td>0.425</td>
</tr>
<tr>
<td>37</td>
<td>2.532</td>
<td>3.611</td>
<td>0.70</td>
<td>0.424</td>
</tr>
<tr>
<td>( n )</td>
<td>( S_n )</td>
<td>( \ln n )</td>
<td>( \frac{S_n}{\ln n} )</td>
<td>( \text{var}(S_n) )</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>38</td>
<td>2.568</td>
<td>3.638</td>
<td>0.71</td>
<td>0.429</td>
</tr>
<tr>
<td>39</td>
<td>2.558</td>
<td>3.664</td>
<td>0.70</td>
<td>0.430</td>
</tr>
<tr>
<td>40</td>
<td>2.592</td>
<td>3.689</td>
<td>0.70</td>
<td>0.434</td>
</tr>
<tr>
<td>41</td>
<td>2.581</td>
<td>3.714</td>
<td>0.70</td>
<td>0.434</td>
</tr>
<tr>
<td>42</td>
<td>2.616</td>
<td>3.738</td>
<td>0.70</td>
<td>0.438</td>
</tr>
<tr>
<td>43</td>
<td>2.605</td>
<td>3.761</td>
<td>0.69</td>
<td>0.440</td>
</tr>
<tr>
<td>44</td>
<td>2.638</td>
<td>3.784</td>
<td>0.70</td>
<td>0.443</td>
</tr>
<tr>
<td>45</td>
<td>2.627</td>
<td>3.807</td>
<td>0.69</td>
<td>0.445</td>
</tr>
<tr>
<td>46</td>
<td>2.659</td>
<td>3.829</td>
<td>0.69</td>
<td>0.447</td>
</tr>
<tr>
<td>47</td>
<td>2.649</td>
<td>3.850</td>
<td>0.69</td>
<td>0.450</td>
</tr>
<tr>
<td>48</td>
<td>2.680</td>
<td>3.871</td>
<td>0.69</td>
<td>0.452</td>
</tr>
<tr>
<td>56</td>
<td>2.754</td>
<td>4.025</td>
<td>0.68</td>
<td>0.469</td>
</tr>
<tr>
<td>64</td>
<td>2.819</td>
<td>4.159</td>
<td>0.68</td>
<td>0.484</td>
</tr>
<tr>
<td>72</td>
<td>2.876</td>
<td>4.277</td>
<td>0.67</td>
<td>0.496</td>
</tr>
<tr>
<td>80</td>
<td>2.927</td>
<td>4.382</td>
<td>0.67</td>
<td>0.506</td>
</tr>
<tr>
<td>88</td>
<td>2.973</td>
<td>4.477</td>
<td>0.66</td>
<td>0.514</td>
</tr>
<tr>
<td>96</td>
<td>3.015</td>
<td>4.564</td>
<td>0.66</td>
<td>0.520</td>
</tr>
<tr>
<td>104</td>
<td>3.053</td>
<td>4.644</td>
<td>0.66</td>
<td>0.526</td>
</tr>
<tr>
<td>112</td>
<td>3.088</td>
<td>4.718</td>
<td>0.65</td>
<td>0.530</td>
</tr>
<tr>
<td>120</td>
<td>3.121</td>
<td>4.787</td>
<td>0.65</td>
<td>0.535</td>
</tr>
<tr>
<td>128</td>
<td>3.151</td>
<td>4.852</td>
<td>0.65</td>
<td>0.538</td>
</tr>
<tr>
<td>136</td>
<td>3.180</td>
<td>4.913</td>
<td>0.65</td>
<td>0.541</td>
</tr>
<tr>
<td>144</td>
<td>3.207</td>
<td>4.970</td>
<td>0.65</td>
<td>0.544</td>
</tr>
<tr>
<td>152</td>
<td>3.232</td>
<td>5.024</td>
<td>0.64</td>
<td>0.547</td>
</tr>
<tr>
<td>n</td>
<td>$S_n$</td>
<td>$\ln n$</td>
<td>$\frac{S_n}{\ln n}$</td>
<td>$\text{var}(S_n)$</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>---------</td>
<td>----------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>160</td>
<td>3.256</td>
<td>5.075</td>
<td>0.64</td>
<td>0.550</td>
</tr>
<tr>
<td>168</td>
<td>3.279</td>
<td>5.124</td>
<td>0.64</td>
<td>0.552</td>
</tr>
<tr>
<td>176</td>
<td>3.300</td>
<td>5.170</td>
<td>0.64</td>
<td>0.555</td>
</tr>
<tr>
<td>184</td>
<td>3.321</td>
<td>5.215</td>
<td>0.64</td>
<td>0.557</td>
</tr>
<tr>
<td>192</td>
<td>3.341</td>
<td>5.257</td>
<td>0.64</td>
<td>0.560</td>
</tr>
<tr>
<td>200</td>
<td>3.360</td>
<td>5.298</td>
<td>0.63</td>
<td>0.562</td>
</tr>
<tr>
<td>225</td>
<td>3.407</td>
<td>5.416</td>
<td>0.63</td>
<td>0.576</td>
</tr>
<tr>
<td>250</td>
<td>3.464</td>
<td>5.521</td>
<td>0.63</td>
<td>0.576</td>
</tr>
<tr>
<td>275</td>
<td>3.501</td>
<td>5.617</td>
<td>0.62</td>
<td>0.589</td>
</tr>
<tr>
<td>300</td>
<td>3.548</td>
<td>5.704</td>
<td>0.62</td>
<td>0.589</td>
</tr>
<tr>
<td>325</td>
<td>3.580</td>
<td>5.784</td>
<td>0.62</td>
<td>0.600</td>
</tr>
<tr>
<td>350</td>
<td>3.620</td>
<td>5.858</td>
<td>0.62</td>
<td>0.600</td>
</tr>
<tr>
<td>375</td>
<td>3.647</td>
<td>5.927</td>
<td>0.62</td>
<td>0.610</td>
</tr>
<tr>
<td>400</td>
<td>3.683</td>
<td>5.991</td>
<td>0.61</td>
<td>0.609</td>
</tr>
<tr>
<td>425</td>
<td>3.706</td>
<td>6.052</td>
<td>0.61</td>
<td>0.619</td>
</tr>
<tr>
<td>450</td>
<td>3.737</td>
<td>6.109</td>
<td>0.61</td>
<td>0.617</td>
</tr>
<tr>
<td>475</td>
<td>3.758</td>
<td>6.163</td>
<td>0.61</td>
<td>0.626</td>
</tr>
<tr>
<td>500</td>
<td>3.786</td>
<td>6.215</td>
<td>0.61</td>
<td>0.624</td>
</tr>
<tr>
<td>525</td>
<td>3.805</td>
<td>6.263</td>
<td>0.61</td>
<td>0.632</td>
</tr>
<tr>
<td>550</td>
<td>3.831</td>
<td>6.310</td>
<td>0.61</td>
<td>0.629</td>
</tr>
<tr>
<td>575</td>
<td>3.848</td>
<td>6.354</td>
<td>0.61</td>
<td>0.637</td>
</tr>
<tr>
<td>600</td>
<td>3.871</td>
<td>6.397</td>
<td>0.61</td>
<td>0.634</td>
</tr>
<tr>
<td>625</td>
<td>3.886</td>
<td>6.438</td>
<td>0.60</td>
<td>0.641</td>
</tr>
<tr>
<td>650</td>
<td>3.908</td>
<td>6.477</td>
<td>0.60</td>
<td>0.638</td>
</tr>
<tr>
<td>$n$</td>
<td>$S_n$</td>
<td>$\ln n$</td>
<td>$\frac{S_n}{\ln n}$</td>
<td>$\text{var}(S_n)$</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>675</td>
<td>3.922</td>
<td>6.515</td>
<td>0.60</td>
<td>0.645</td>
</tr>
<tr>
<td>700</td>
<td>3.942</td>
<td>6.551</td>
<td>0.60</td>
<td>0.642</td>
</tr>
<tr>
<td>725</td>
<td>3.955</td>
<td>6.586</td>
<td>0.60</td>
<td>0.648</td>
</tr>
<tr>
<td>750</td>
<td>3.974</td>
<td>6.620</td>
<td>0.60</td>
<td>0.645</td>
</tr>
<tr>
<td>775</td>
<td>3.986</td>
<td>6.653</td>
<td>0.60</td>
<td>0.651</td>
</tr>
<tr>
<td>800</td>
<td>4.003</td>
<td>6.685</td>
<td>0.60</td>
<td>0.648</td>
</tr>
<tr>
<td>825</td>
<td>4.014</td>
<td>6.715</td>
<td>0.60</td>
<td>0.654</td>
</tr>
<tr>
<td>850</td>
<td>4.031</td>
<td>6.745</td>
<td>0.60</td>
<td>0.651</td>
</tr>
<tr>
<td>875</td>
<td>4.041</td>
<td>6.774</td>
<td>0.60</td>
<td>0.656</td>
</tr>
<tr>
<td>900</td>
<td>4.057</td>
<td>6.802</td>
<td>0.60</td>
<td>0.653</td>
</tr>
<tr>
<td>925</td>
<td>4.067</td>
<td>6.830</td>
<td>0.60</td>
<td>0.658</td>
</tr>
<tr>
<td>950</td>
<td>4.082</td>
<td>6.856</td>
<td>0.60</td>
<td>0.655</td>
</tr>
<tr>
<td>975</td>
<td>4.091</td>
<td>6.882</td>
<td>0.59</td>
<td>0.660</td>
</tr>
<tr>
<td>1000</td>
<td>4.105</td>
<td>6.908</td>
<td>0.59</td>
<td>0.657</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


