JOURNAL OF ALGORITHMS 9, 129–136 (1988)

A New Proof of the Garsia–Wachs Algorithm

JEFFREY H. KINGSTON

Computer Science Department, The University of Iowa, Iowa City, Iowa 52242

Received November 24, 1986

A new proof of correctness for the Garsia–Wachs algorithm is presented. Like the well-known Hu–Tucker algorithm, the Garsia–Wachs algorithm constructs minimum cost binary trees in $O(n \log n)$ time. The new proof has a simpler structure and is free of the difficult inductions of the original. © 1988 Academic Press, Inc.

1. INTRODUCTION

We are concerned with the class of binary trees with nonnegative real numbers, or *weights*, attached to their leaves. The problem is to find, for a given list of n weights, the tree with minimum weighted external path length. Unlike the problem solved by the Huffman algorithm [5], where the weights may appear in any order, we require the weights to appear, from left to right, in the order given.

The first algorithm for this task was presented in the seminal paper of Gilbert and Moore [2]. Their algorithm was improved from $O(n^3)$ to $O(n^2)$ by Knuth [6]. His algorithm solves a more general problem, in which the internal nodes may also have weights.

In 1971, the radically new Hu–Tucker algorithm appeared [3]. It solved the problem in $O(n \log n)$ time, but its proof was very long and difficult. Hu later gave a shorter proof [4].

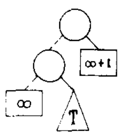
The next step was taken by Garsia and Wachs [1]. They altered the algorithm substantially and were thereby able to produce a shorter and clearer proof of correctness. Their algorithm is described in Section 3 below.

These last papers have greatly simplified the presentation of the Hu–Tucker algorithm, but it remains difficult to comprehend. We believe that our proof of the Garsia–Wachs algorithm is the first to permit an intuitive understanding of why this class of algorithms is correct.

0196-6774/88 $3.00

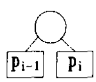
2. Definitions

We will closely follow the terminology of Garsia and Wachs, with two exceptions: first, we revert to diagrams of trees rather than using their tree language; and second, along with the weights $p_1, \dots, p_n$ we include two "sentinel" weights $p_0 = \infty$ and $p_{n+1} = \infty + 1$, where ∞ is any number greater than $\sum_{i=1}^{n} p_i$. By Theorem 5 of Gilbert and Moore [2], if T is the desired minimal tree, then the minimal tree for the extended list of weights is



and so this extension conveniently removes the boundary conditions of our theorems.

Definition 1. A (binary) tree T is either a positive real number p_i, or else it has a root and two binary subtrees. If $T_1, \dots, T_k$ are trees, we denote by $\mathscr{T}(T_1, \dots, T_k)$ the set of all trees with k leaves, but with those leaves replaced by $T_1, \dots, T_k$ from left to right. For a fixed i, we let

T_x =


and define

$$\mathscr{T} = \mathscr{T}(p_0, p_1, \dots, p_n, p_{n+1})$$
$$\mathscr{T}_0 = \mathscr{T}(p_0, p_1, \dots, p_{i-2}, T_x, p_{i+1}, \dots, p_n, p_{n+1})$$
$$\mathscr{T}_k = \mathscr{T}(p_0, p_1, \dots, p_{i-2}, p_{i+1}, \dots, p_{i+k}, T_x, p_{i+k+1}, \dots, p_n, p_{n+1}).$$

Definition 2. The *level*, h_i, of any leaf p_i is the number of internal nodes on the path from the root to p_i (or equivalently, the number of arcs). We will extend this definition slightly, by speaking of the root of T_x as node p_x of level h_x ($p_x = p_{i-1} + p_i$). The weight of $T \in \mathscr{T}$, $\mathscr{T}_0$, or $\mathscr{T}_k$ is

$$w(T) = \sum_{i=0}^{n+1} h_i p_i,$$

and we define

$$w(S) = \min_{T \in S} w(T)$$

for any set S of trees. Finally, a tree $T \in S$ will be called *minimal* for S if $w(T) = w(S)$.

DEFINITION 3. Let $U \in \mathscr{T}(p_0, p_1, \ldots, p_n, p_{n+1})$ and $V \in \mathscr{T}(q_0, q_1, \ldots, q_n, q_{n+1})$, where $p_0 = q_0 = \infty$ and $p_{n+1} = q_{n+1} = \infty + 1$. Let h_i be the level of p_i in U, and k_i be the level of q_i in V. We say that U is a *rearrangement* of V (briefly, $U \sim V$), if there is a permutation $(\sigma_1, \ldots, \sigma_n)$ of $(1, 2, \ldots, n)$ such that $p_i = q_{\sigma_i}, h_i = k_{\sigma_i}$ for $1 \leq i \leq n$. Informally, a rearrangement merely moves leaves around within the tree, without altering their level. Consequently, " $\sim$ " is an equivalence relation, and $U \sim V$ implies $w(U) = w(V)$.

DEFINITION 4. A pair of leaves p_{i-1}, p_i is *right minimal* (briefly, R.M.) if

(i) $1 < i \leq n$ (sentinel nodes do not participate)

(ii) $p_{i-2} + p_{i-1} \geq p_{i-1} + p_i$

(iii) $p_{i-1} + p_i < p_{j-1} + p_j$ for all $j > i$.

3. THE GARSIA–WACHS ALGORITHM

The algorithm constructs a rearrangement T_B of the minimal tree T. Once this is done, the levels of the p_i in T_B may be used to construct T. Beginning with the list of weights

$$p_0, p_1, \ldots, p_n, p_{n+1}$$

the algorithm executes the following two steps $n - 1$ times:

(i) Locate the rightmost R.M. pair of entries. Let that be p_{i-1}, p_i.

(ii) Next locate the first entry to the right of p_i that is greater than or equal to $p_{i-1} + p_i$. Let that be p_{i+k+1}. Then the new list is

$$p_0, p_1, \ldots, p_{i-2}, p_{i+1}, \ldots, p_{i+k}, (p_{i-1} + p_i), p_{i+k+1}, \ldots, p_n, p_{n+1}.$$

Each sum $p_{i-1} + p_i$ represents an internal node p_x of T_B.

4. PROOF OF CORRECTNESS

The theorem we prove is somewhat weaker than the corresponding one of Garsia and Wachs, in that we restrict our attention to the rightmost R.M. pair.

THEOREM 1. *Let p_{i-1}, p_i be the rightmost R.M. pair, and let $k \geq 0$ be such that*

$$p_{i+j} < p_{i-1} + p_i \qquad \text{for } 0 \leq j \leq k$$

and

$$p_{i+k+1} \geq p_{i-1} + p_i.$$

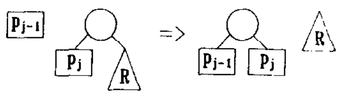
Then $w(\mathcal{T}) = w(\mathcal{T}_k)$, and every minimal tree for $\mathcal{T}_k$ has a rearrangement in $\mathcal{T}$.

LEMMA 1. *Suppose we have a sequence of at least three nodes $p_a, p_{a+1}, \ldots, p_b$ such that*

$$p_{j-1} + p_j < p_j + p_{j+1} \qquad \text{for } a < j < b.$$

Then $h_a \geq h_{a+1} \geq \cdots \geq h_{b-1}$ in every minimal tree containing $p_a, \ldots, p_b$.

Proof. Suppose $h_{j-1} < h_j$ for some j such that $a < j < b$. Then p_j is a left child, and the transformation



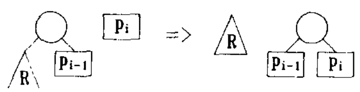
produces a better tree, since $|R| \geq p_{j+1} > p_{j-1}$ by assumption (here $|R|$ is the total weight of R).

LEMMA 2. *If p_{i-1}, p_i is the rightmost R.M. pair, then $h_{i-1} \geq h_i \geq \cdots \geq h_n$ in every minimal tree.*

Proof. We must have $p_{j-1} + p_j < p_j + p_{j+1}$, for all $j \geq i$, since otherwise either the pair p_j, p_{j+1} would be R.M., or some smaller pair to its right would be. Hence, the sequence $p_{i-1}, p_i, \ldots, p_{n+1}$ satisfies the conditions of Lemma 1.

LEMMA 3. *If p_{i-1}, p_i is the rightmost R.M. pair, then $h_{i-1} = h_i$ in some minimal tree.*

Proof. By Lemma 2, we need only show $h_{i-1} \leq h_i$ in some minimal tree. But suppose $h_{i-1} > h_i$. Then p_{i-1} is a right child, and the transformation



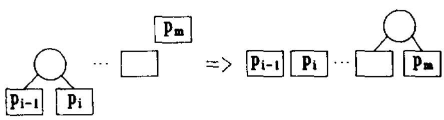
yields an equal or better tree, since $|R| \geq p_{i-2} \geq p_i$ by the right minimality of p_{i-1}, p_i. So this new tree is also minimal, and $h_{i-1} = h_i$ in it.

LEMMA 4. *Let p_{i+k+1} be the first node to the right of the rightmost R.M. pair p_{i-1}, p_i such that $p_{i+k+1} \geq p_{i-1} + p_i$. Then in some minimal tree T for which $h_{i-1} = h_i$, either*

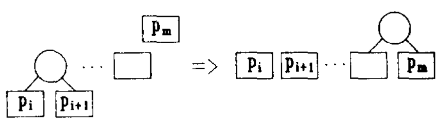
(a) $h_{i+k} = h_i - 1$; *or*

(b) $h_{i+k} = h_i$ *and* p_{i+k} *is a right child.*

Proof. We first show that $h_{i+k} = h_i - 1$ or $h_{i+k} = h_i$. Begin with the minimal tree T of Lemma 3. By Lemma 2, it is sufficient to show that $h_{i+k} \geq h_i - 1$ in T. So suppose on the contrary that $h_{i+k} < h_i - 1$. Let p_m be the first node to the right of p_i such that $h_m < h_i - 1$. By hypothesis, $m \leq i + k$ and so $p_m < p_{i-1} + p_i$. Now consider the transformations

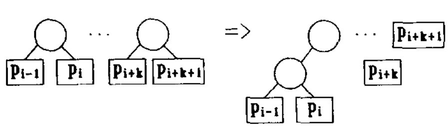


or

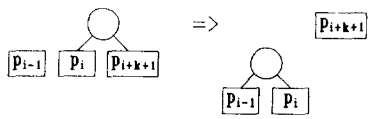


chosen depending on whether p_{i-1} and p_i are siblings in T. The first gives a decrease in weight of $p_{i-1} + p_i - p_m > 0$; the second gives a decrease of $p_i + p_{i+1} - p_m > p_{i-1} + p_i - p_m > 0$. This contradicts the minimality of T, so we must have $h_{i+k} \geq h_i - 1$.

Now suppose that $h_{i+k} = h_i$. If p_{i+k} is a right child, we are done. If p_{i+k} is a left child and $k > 0$, the transformation



shows that a minimal tree of type (a) exists, while if $k = 0$ the transformation



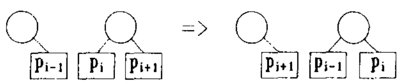
shows that a minimal tree of type (b) exists.

We have accumulated some interesting facts about a minimal tree T, with little effort. For our final two lemmas we must enter the more difficult world of the rearrangements of T.

LEMMA 5. $w(\mathscr{T}_k) \le w(\mathscr{T})$.

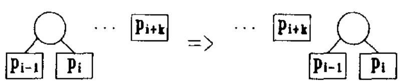
Proof. This will follow if we can exhibit a tree $T' \in \mathscr{T}_k$ such that $T' \sim T$, where T is the minimal tree of Lemma 4, for then $w(\mathscr{T}_k) \le w(T') = w(T) = w(\mathscr{T})$. If $k = 0$, we may take $T' = T$; otherwise we proceed as follows.

The first step in constructing T' is to rearrange T so that p_{i-1} and p_i are siblings (if they are not so already):

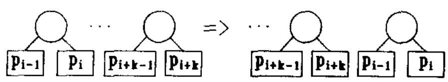


Since $k > 0$, p_{i+k+1} is still somewhere to the right of p_i after this step.

The second step is to take the tree T_x so formed and move it to the right until it passes over p_{i+k}:



or

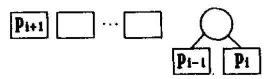


These are the only two possibilities, by Lemma 4, and the result in each case is a suitable T'.

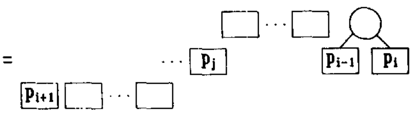
LEMMA 6. *Let T_k be any minimal tree for $\mathscr{T}_k$. Then $w(T_k) \ge w(\mathscr{T})$, and if equality holds, T_k has a rearrangement in $\mathscr{T}$.*

Proof. Conversely to Lemma 5, we must now find a tree $T \in \mathcal{T}$ such that $w(T) \leq w(T_k)$. If $k = 0$, this is immediate by letting $T = T_k$, since $\mathcal{T}_0 \subset \mathcal{T}$, and clearly T_k has a rearrangement (namely, itself) in $\mathcal{T}$.

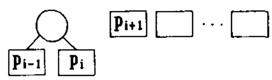
Otherwise, we know that $p_{i+k} < p_x \leq p_{i+k+1}$ in T_k, and so we may take $p_a, \ldots, p_b = p_{i+1}, \ldots, p_{i+k}, p_x, p_{i+k+1}$ in the statement of Lemma 1 to obtain $h_{i+1} \geq h_{i+2} \geq \cdots \geq h_{i+k} \geq h_x$ in T_k. Hence,

$T_k =$ 

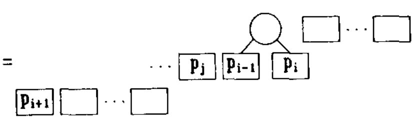
or

$T_k =$ 

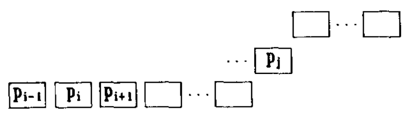
In the first case, it is a simple matter to move T_x to the left to obtain

$T =$ 

and we see that $T \in \mathcal{T}$ and $T \sim T_k$, so the lemma is proved. In the second case, we first move T_x to the left across all leaves of level h_x to obtain the rearrangement

$T^* =$ 

Next, we *rotate* all the nodes $p_{i+1}, \ldots, p_j, p_{i-1}, p_i$ two places to the right, without disturbing the shape of T^*, giving T:

$T =$ 

For every level the pair p_{i-1}, p_i drop downward (possibly 0), some other pair p_{s-1}, p_s move up one level. But $p_{s-1} + p_s > p_{i-1} + p_i$, since p_{i-1}, p_i were R.M., so we must have $w(T_k) = w(T^*) \geq w(T)$, and equality implies that the level of p_{i-1}, p_i is not changed by the rotation, i.e., that $T_k \sim T$.

Theorem 1 now follows immediately from Lemmas 5 and 6.

ACKNOWLEDGMENT

The author thanks the referee for a substantial correction and simplification.

REFERENCES

1. A. M. GARSIA AND M. L. WACHS, A new algorithm for minimum cost binary trees, *SIAM J. Comput.* **6** (1977), 622–642.
2. E. N. GILBERT AND E. F. MOORE, Variable length binary encodings, *Bell Systems Tech. J.* **38** (1959), 933–968.
3. T. C. HU AND C. TUCKER, Optimum computer search trees, *SIAM J. Appl. Math.* **21** (1971), 514–532.
4. T. C. HU, A new proof of the T–C algorithm, *SIAM J. Appl. Math.* **25** (1973), 83–94.
5. D. HUFFMAN, A method for the construction of minimum-redundancy codes, *Proc. IRE* **40** (1952), 1098–1101.
6. D. E. KNUTH, Optimum binary search trees, *Acta Inform.* **1** (1971), 14–25.