AN INFORMATION-THEORETIC LOWER BOUND FOR THE LONGEST COMMON SUBSEQUENCE PROBLEM *

D.S. HIRSCHBERG
Rice University, Houston TX 77001

Received 1 July 1977; revised version received 26 October 1977

Algorithm, comparison

The longest common subsequence (LCS) problem is the problem of determining a sequence C of maximum length that is a subsequence of (can be obtained by deleting zero or more symbols from) each of two given strings A and B [1].

The best algorithms known for the LCS problem are, in the worst case, only slightly faster than quadratic in the length of the input [3,5] although, for some special cases, there are algorithms known that require only $O(n \log n)$ time [3,4].

Lower bounds on the complexity of the LCS problem have been determined for algorithms that are restricted to making "equal–unequal" comparisons of positions in the two strings. A "comparison of two positions" means a comparison of the values of the symbols located at those positions. It has been shown [1] that $O(n^2)$ such comparisons are required to solve the LCS problem for unrestricted alphabet size and $O(ns)$ such comparisons are required for alphabet size restricted to s.

We shall prove that $n \log n$ is a lower bound on the number of "less than-equal-greater than" comparisons required to solve the LCS problem, assuming unrestricted alphabet size.

Let $T(n)$ be the minimum number of comparisons (resulting in "less than", "greater than", of "equal") required to solve the LCS problem with two input strings of length n.

We shall use a decision tree model (see [1]) and shall demonstrate a lower bound on $T(n)$ by exhibiting a path of sufficient length in each possible decision tree.

A *basic configuration* is an assignment of values to strings A and B such that there are no values common to strings A and B. Thus a basic configuration has an LCS of length 0.

A *valid configuration* (for a particular sequence of comparisons) is an assignment of values to positions that is consistent with the results of all comparisons.

We now define an "oracle" or decision rule by which a path, P_*, is distinguished in each decision tree for the LCS problem. Let $P_*^{(i)}$ be the prefix of length i of P_* (starting at the root of the decision tree).

Decision rule. Let the comparison $p_1 : p_2$ be the ith on P_*. If p_1 and p_2 are both positions in A (say, a_u and a_v) then if $u < v$ then return "less than"; otherwise, return "greater than".

If p_1 and p_2 are not both positions in A then do the following. Let R be the set of relative orderings of the positions of strings A and B that are consistent with the results of all comparisons made along $P_*^{(i-1)}$ that also have $a_1 < a_2 < \cdots < a_n$. Let R_1 be the subset of R that is consistent with $p_1 < p_2$ and let R_2 be the subset of R consistent with $p_1 > p_2$. If $|R_1| > |R_2|$ then return "less than"; otherwise return "greater than". □

Note that the decision rule never returns a result of "equal".

Define positions p and q to be *comparable* (for a sequence of comparisons) if it can be logically deduced

* Research supported by NSF Grant MCS-76-07683.

from the results of the comparisons that $p < q$ or that $p > q$.

Lemma. *There must be sufficient comparisons in P_* so that all positions in A are comparable (possibly by transitivity) to all positions in B.*

Proof. If not, assume a_i is not comparable to b_j. We know that there is a valid basic configuration C_* for P_* in which $a_i < b_j$ and which has an LCS of length 0.

Consider the set S of positions p (of A and/or B) in C_* such that $a_i < p < b_j$. We can partition S into subsets S_0, S_1, S_2, and S_3 where

$S_0 = \{p_0 \in S \mid p_0$ not comparable to either a_i or $b_j\}$,
$S_1 = \{p_1 \in S \mid p_1$ comparable to a_i but not to $b_j\}$,
$S_2 = \{p_2 \in S \mid p_2$ comparable to b_j but not to $a_i\}$,
$S_3 = \{p_3 \in S \mid p_3$ comparable to both a_i and $b_j\}$.

In what follows, p is a generic element of S, p_k is a generic element of S_k (for $k = 0, 1, 2, 3$). S_3 is empty since otherwise a_i is comparable to b_j. There is no $p_1 \in S_1$ that is comparably less than any $p_2 \in S_2$ since otherwise a_i would be comparably less than b_j. Also, there is no $p_0 \in S_0$ that is comparably greater than any $p_1 \in S_1$ or is comparably less than any $p_2 \in S_2$ since otherwise p_0 would be in S_1 or S_2 respectively.

We can change the relative order of values of a_i, $\{p\}$, b_j so that $\{p_2\} < a_i < b_j < \{p_0\} < \{p_1\}$ and will still have a valid basic configuration C_0. The configuration, C_1, which is the same as C_0 except that $a_i = b_j$ will also be valid, but it will have an LCS of length 1. The decision tree D, of which P_* was a path, does not distinguish between these two valid configurations and hence does not solve the LCS problem. □

Lemma. *There must be $n \log n$ comparisons along P_*.*

Proof. Each element, b_j of B, can be in any one of $n + 1$ distinct states:

$b_j \leqslant a_1$,
$a_i < b_j \leqslant a_{i+1}$, $[i = 1, ..., n - 1]$,
$a_n < b_j$.

Thus, there are $(n + 1)^n$ possible relative orderings of the elements of B with respect to the elements of A. It will require $\log((n + 1)^n) \geqslant n \log n$ comparisons to distinguish which states the elements of B are in. That is, $n \log n$ comparisons are required to make every element of B comparable to every element of A. □

Theorem. $T(n) \geqslant n \log n$.

Proof. We have exhibited a path of length $n \log n$ that must appear in any decision tree that solves the LCS problem. □

References

[1] A.V. Aho, D.S. Hirschberg and J.D. Ullman, Bounds on the complexity of the longest common subsequence problem, J. ACM 23 (1) (January 1976) 1–12.
[2] D.S. Hirschberg, A linear space algorithm for computing maximal common subsequences, Comm. ACM 18 (6) (June 1975) 341–343.
[3] D.S. Hirschberg, Algorithms for the longest common subsequence problem, J. ACM 24 (3) (October 1977).
[4] J.W. Hunt and T.G. Szymanski, A fast algorithm for computing longest common subsequences, Comm. ACM 20 (5) (May 1977) 350–353.
[5] M.S. Paterson, unpublished manuscript, University of Warwick, England (1974).
[6] C.K. Wong, private communication to D.S. Hirschberg.