AN INFORMATION THEORETIC LOWER BOUND FOR
THE LONGEST COMMON SUBSEQUENCE PROBLEM

D.S. Hirschberg
Rice University

April 1977

Rice Technical Report No. 7705

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

The best algorithms known for the LCS problem are quadratic for unrestricted alphabet size [H1] and only slightly faster for restricted alphabet size [H2, Pa].

Lower bounds on the complexity of the LCS problem have been determined for algorithms that are restricted to making "equal-unequal" comparisons. It has been shown [AHiU] 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 derive a lower bound on the number of "less than-equal-greater than" comparisons required to solve the LCS problem.

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

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 is distinguished in each decision tree for the LCS problem.

Decision Rule: Let the comparison $p_1:p_2$ be the i th on P_* . If p_1 and p_2 are both positions in A (say, a_i and a_j) then return "less than" if $i < j$. 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 elements of strings A and B that are consistent with the results of all comparisons made along $P^{(i-1)}$ that also have $a_1 < a_2 < \dots < a_n$. Let R_1 be that subset of R that is consistent with $p_1 < p_2$ and let R_2 be that subset of R consistent with $p_1 > p_2$. If $|R_1| > |R_2|$ then return "less than", otherwise return "greater than".

Define positions p and q to be comparable (for a sequence of comparisons) if it can be logically deduced 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 elements c (of A and/or B) in C^* such that $a_i < c < b_j$. We can partition S into subsets S_0, S_1, S_2 , and S_3 where

$$S_0 = \{c_0 \in S \mid c_0 \text{ not comparable to either } a_i \text{ or } b_j\},$$

$$S_1 = \{c_1 \in S \mid c_1 \text{ comparable to } a_i \text{ but not to } b_j\},$$

$$S_2 = \{c_2 \in S \mid c_2 \text{ comparable to } b_j \text{ but not to } a_i\},$$
 and $S_3 = \{c_3 \in S \mid c_3 \text{ comparable to both } a_i \text{ and } b_j\}.$
 S_3 is empty since otherwise a_i is comparable to b_j . There is no $c_1 \in S_1$ that is comparably less than any $c_2 \in S_2$ since otherwise a_i would be comparably less than b_j . Also, there is no $c_0 \in S_0$ that is comparably greater than any $c_1 \in S_1$ or is comparably less than any $c_2 \in S_2$ since otherwise c_0 would be in S_1 or S_2 respectively.

We can change the relative order of values of $a_i, \{c\}, b_j$ so that $\{c_2\} < a_i < b_j < \{c_0\} < \{c_1\}$ 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. $\square$

Lemma. There must be $O(n \log n)$ comparisons along P^* .

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

$$\begin{array}{l} b_j < a_1 \\ a_i < b_j < a_{i+1} \\ a_n < b_j \end{array} \quad [i=1, \dots, n-1]$$

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

Theorem. $T(n) \geq O(n \log n)$.

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

References

- AH10 Aho A.V., D.S. Hirschberg and J.D. Ullman. Bounds on the Complexity of the Longest Common Subsequence Problem. J.ACM 23:1, January 1976, pp.1-12.
- H1 Hirschberg D.S. A Linear Space Algorithm for Computing Maximal Common Subsequences. Comm. ACM 18:6, June 1975, pp.341-343.
- H2 Hirschberg D.S. Algorithms for the Longest Common Subsequence Problem. J.ACM to appear.
- Pa Paterson M.S. Unpublished manuscript. University of Warwick, England, 1974.
- Wo Wong C.K. private communication to D.S. Hirschberg.