Longest Decreasing Subsequence

ZPRAC-16-17-Lab11

Longest Decreasing Subsequence [40 points]

ANNOUNCEMENT:
Up to 20% marks will be allotted for good programming practice. These include
- Comments for non trivial code
- Indentation: align your code properly

Given a sequence of N integers, you must find the length of the longest decreasing subsequence.

A subsequence is obtained by deleting zero or more elements from a sequence. A decreasing (or “non-increasing”) subsequence is one in which the elements are in decreasing order with repetition of equal elements allowed.

For example, consider the sequence {18,5,20,2,2,10,0}. The LDS (longest decreasing subsequence) is {18,5,2,2,0} of length 4.

One can compute the LDS of a sequence using the following observation:

Let LDS[n], 1 ≤ n ≤ N, denote the length of the longest decreasing subsequence with A[n] as the last element of the subsequence. Using LDS[i], for an i < n, the subsequence obtained by adding A[n] (provided A[n] is less than or equal to A[i]) will have +1 length. Hence, LDS[i] can be computed as follows:

$$LIS[n]=1+MAX_{i=1,2,...n-1 \text{ and } A[n]\leq A[i]}(LIS[i])$$

The length of the longest increasing subsequence can then be computed by finding the maximum of LIS[n] for all possible values of n.

Input Format:
First line contains an integer N denoting the length of the sequence.
The next line contains N space separated integers denoting the contents of the sequence.

Constraints:
1 ≤ N ≤ 1000

Output Format:
A single integer, which is the length of the longest decreasing subsequence.

Examples:

Given Input:
7
18 5 20 2 2 10 0

Expected Output:
5

Explanation: The subsequence {18, 5, 2, 2, 0} is the longest decreasing subsequence with length 5.