Longest Common Substring

ZPRAC-16-17-Lab11

Longest Common Substring [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 two strings A and B of length N and M, you must find their longest common substring.

A substring is defined as any continuous sequence of characters of a string.

Observe that the brute force solution of generating all possible substrings of one string and checking it with the other is inefficient. However, one can design an efficient solution using the following observation:

Let LCS[m][n] denote the length of the longest substring ending at index 'm' of A and index 'n' of B, $1 \leq n \leq N$, $1 \leq m \leq M$. If we know the value of LCS[m-1][n-1], then we can compute LCS[m][n] as follows:

$LCS[m][n]=1+LCS[m-1][n-1]$ if m^{th} character A is same as n^{th} character of B

$LCS[m][n]=0$ otherwise

Using this dependence of LCS[m][n], one can find the longest common substring of A and B by computing LCS for every m and n, and finding the maximum value of LCS over these possibilities.

Input Format:
First line contains two integers N and M denoting the length of the first and second string.
Second line contains the contents of the first string.
Third line contains the contents of the second string.

Constraints:
$1 \leq N \leq 1000$
$1 \leq M \leq 1000$

Output Format:
A single integer, which is the length of the longest common substring of the two given strings.

Examples:

Given Input:
6 5
plural
rural

Expected Output:
4