Question 1. No. Counterexample: $X = aab$, $Y = baa$, $D[a] = 1$, $D[b] = 3$. The algorithm erroneously produces a cost of 6 (deletion of the two b symbols) when a cost of 4 is possible (deletion of the four a symbols).

Question 2. $c_{i,j} = \min\{c_{i-1,j-1} + S[x_i, y_j], c_{i-1,j} + D[x_i], c_{i,j-1} + I[y_j]\}$.

Question 3. Give the software as input the two sequences whose LCS we seek, an I and a D vector all of whose entries are 1, and an S matrix whose diagonal entries are 0 and all its other entries are large to ensure that substitutions are too expensive to be used (3 is large enough for the off-diagonal values of S). Because no substitutions are used, and the cost of every insertion or deletion is 1, an optimal solution computed by the software corresponds to deleting from X all the $m - \ell$ entries that are not part of the computed LCS of X and Y, and inserting in X all the $n - \ell$ entries of Y that are not in that LCS. Therefore the cost $c_{m,n}$ computed by the software is the sum of the above-mentioned deletion costs ($= m - \ell$) and insertion costs ($= n - \ell$):

Assignment Project Exam Help

$$c_{m,n} = (m - \ell) + (n - \ell)$$

giving a value of ℓ equal to $(m + n - c_{m,n})/2$.

https://tutorcs.com

Question 4. Create B = a sorted version of A (by increasing values). Then give A and B as input to the LCS software: The returned LCS is an LIS of A; it is increasing because B itself is sorted, and it is of maximum possible length because an LCS has (by definition) maximum length.

WeChat: cstutorcs