**LONGEST COMMON SUB\_SEQUENCE**

**Correctness and Time Complexity**

The LCS problem has an optimal substructure.That means the problem can be broken down into smaller ,simple “subproblem”,which can be broken down into yet simpler subproblems,, and so on, until,the solution become trivial.

1. Let us consider two sequences X and Y of length m and n that both ends in the same element

To find their LCS ,shorten each sequence by removing the last element,find the LCS of the shortened sequences ,and that LCS append the removed elements.So, we can say that

LCS(X[1..m],Y[1..n]=LCS(X[1…m-1],Y[1..n-1])+X[m] if X[m] =Y[n]

1. Now suppose that the two sequences do not end in the same symbol.

Then the LCS of X and Y is the longer of the two sequences LCS(X[1…m-1],Y[1…n]) and LCS(X[1…m],Y[1…n-1]).To understand this property,lets consider the following two sequences.

X=ABCBDAB(n elements)

Y=BDCABA(m elements)

The LCS of these two sequences ends with a B.

CASE 1: If LCS ends with a B,then it cannot end with a A and we can remove the A from sequence Y and then problem reduces to LCS(X[1…m], Y[1…n-1].

CASE 2: If LCS does not ends with a B,then we can remove the B from sequence X and then problem reduces to LCS(X[1…m-1], Y[1…n].

The **worst case time complexity** of LCS is O(2^(m+n)).The worst case happen when there is no common subsequence present in X and Y and each recursive call will end up in two recursive calls.

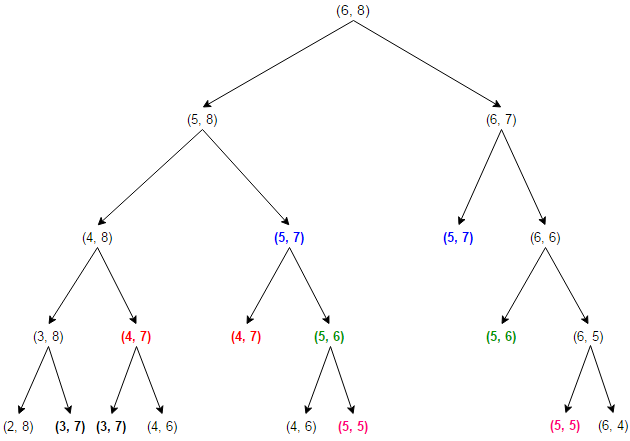
For Example

X=ABCD

Y=EFGH

The LCS problem exhibit overlapping sub problems. A problem is said to have overlapping sub problem if the recursive algorithm for the problem solve the same sub problem over and over rather than generating the same sub problem.

Let us consider recursion tree of two sequence of length 6 and 8 whose LCS is 0.



As we can see in fig of tree ,the same sub problem are getting computed again and again .We know that problem having optimal substructure and overlapping subproblem can be solved by dynamic ,in which sub problem solution are memorized rather than computed again and again.

The **best case time complexity** of LCS is O(n.m).The best case happen when all the elements in the X and Y are common subsequence.

For Example

X=ABCD

Y=ABCD

The **Space complexity** is O(n\*m).

The memorized version follow the top down approach,since we first break the problem into subproblems and then calculate and store values.We can also save it in bottom up manner.In the bottom up approach,we calculate the smaller value of LCS(i,j) first then builder values from them.

Let X be XMJYAUZ and Y be MZJAWXU .The longest common sub sequence between X and Y is MJAU. The table below is generated by the function of LCSLenghth ,shows the length of the longest common sub sequence between prefixes of X and Y.The ith row and jth column shows the length of the LCSof substring X[0…i-1]and Y[0..j-1].

