Minimum number of insertions or deletions to make it a palindrome.

1)lets take a string s1= “ACDBA”. Now to find a palindrome make reverse of string s1

2)Find LCS (s1, reverse of string s1).Then we get common subsequence and the remaining letter are not required so we need to delete those.

3) No of deletions required = S1. length () – LCS

CODE:

public class NumOfDeletions {

public static void main(String args[])

{

String s1="BRTSSCB";

StringBuilder sb=new StringBuilder(s1);

sb=sb.reverse();

String s2=new String(sb);

int LCSlength=LCS(s1,s2);

int NumOfDeletions=s1.length()-LCSlength;

System.out.println(NumOfDeletions);

}

public static int LCS(String s1,String s2)

{

int n1=s1.length();

int n2=s2.length();

int[][] dp=new int[n1+1][n2+1];

for(int i=0 ;i <n1+1 ;i++)

{

for(int j=0; j<n2+1 ;j++)

{

if(i==0 || j==0 )

{

dp[i][j]=0;

}

else if(s1.charAt(i-1)==s2.charAt(j-1))

{

dp[i][j]=1+dp[i-1][j-1];

}

else

{

dp[i][j]=Math.max(dp[i][j-1],dp[i-1][j]);

}

}

}

return dp[n1][n2];

}

}