

模版·动态规划(最长公共子序列)

```
public class LongestCommonSubsequence{

    public static String lcsString(String text1,String text2){
        int m=text1.length();
        int n=text2.length();

        int[][] dp=new int[m+1][n+1];
        for(int i=1;i<=m;i++){
            for(int j=1;j<=n;j++){
                if(text1.charAt(i-1)==text2.charAt(j-1)){
                    dp[i][j]=dp[i-1][j-1]+1;
                }else{
                    dp[i][j]=Math.max(dp[i-1][j],dp[i][j-1]);
                }
            }
        }
        StringBuilder lcs=new StringBuilder();
        int i=m,j=n;
        while(i>0&& j>0){
            if(text1.charAt(i-1)==text2.charAt(j-1)){
                lcs.append(text1.charAt(i-1));
                i--;
                j--;
            }else if(dp[i-1][j]>dp[i][j-1]){
                i--;
            }else{
                j--;
            }
        }
    }
}
```

```
        return lcs.reverse().toString();
    }
    public static void main(String[] args){
        String text1="ABCB DAB";
        String text2="BDCAB";
        System.out.println("LCSstring:" + lcsString(text1,text2));
    }
}
```