### 实验五：最长公共子序列（LCS）

1）实验内容

1. 编程实现最长公共子序列（LCS）算法，并理解其核心思想。

2. 时间复杂度O(mn)，空间复杂度O(mn)，求出LCS及其长度。

3. 时间复杂度O(mn)，空间复杂度O(2\*min(m,n))，求出LCS的长度。

4. 时间复杂度O(mn)，空间复杂度O(min(m,n))，求出LCS的长度。

2）算法设计思路

最长公共子序列的结构有如下表示：

设序列X=<x1, x2, …, xm>和Y=<y1, y2, …, yn>的一个最长公共子序列Z=<z1, z2, …, zk>，则：

若xm=yn，则zk=xm=yn且Zk-1是Xm-1和Yn-1的最长公共子序列；

若xm≠yn且zk≠xm ，则Z是Xm-1和Y的最长公共子序列；

若xm≠yn且zk≠yn ，则Z是X和Yn-1的最长公共子序列。

其中Xm-1=<x1, x2, …, xm-1>，Yn-1=<y1, y2, …, yn-1>，Zk-1=<z1, z2, …, zk-1>。

3、2.子问题的递归结构

由最长公共子序列问题的最优子结构性质可知，要找出X=<x1, x2, …, xm>和Y=<y1, y2, …, yn>的最长公共子序列，可按以下方式递归地进行：当xm=yn时，找出Xm-1和Yn-1的最长公共子序列，然后在其尾部加上xm(=yn)即可得X和Y的一个最长公共子序列。当xm≠yn时，必须解两个子问题，即找出Xm-1和Y的一个最长公共子序列及X和Yn-1的一个最长公共子序列。这两个公共子序列中较长者即为X和Y的一个最长公共子序列。

由此递归结构容易看到最长公共子序列问题具有子问题重叠性质。例如，在计算X和Y的最长公共子序列时，可能要计算出X和Yn-1及Xm-1和Y的最长公共子序列。而这两个子问题都包含一个公共子问题，即计算Xm-1和Yn-1的最长公共子序列。

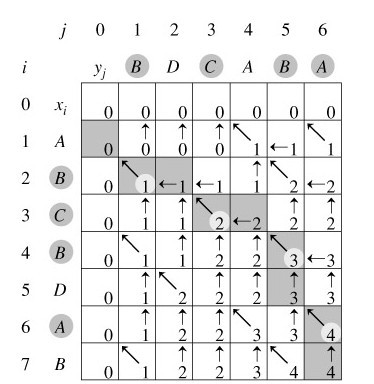
由算法LCS\_LENGTH计算得到的数组b可用于快速构造序列X=<x1, x2, …, xm>和Y=<y1, y2, …, yn>的最长公共子序列。首先从b[m,n]开始，沿着其中的箭头所指的方向在数组b中搜索。

当b[i,j]中遇到"↖"时（意味着xi=yi是LCS的一个元素），表示Xi与Yj的最长公共子序列是由Xi-1与Yj-1的最长公共子序列在尾部加上xi得到的子序列；

当b[i,j]中遇到"↑"时，表示Xi与Yj的最长公共子序列和Xi-1与Yj的最长公共子序列相同；

当b[i,j]中遇到"←"时，表示Xi与Yj的最长公共子序列和Xi与Yj-1的最长公共子序列相同。

这种方法是按照反序来找LCS的每一个元素的。由于每个数组单元的计算耗费Ο(1)时间，算法LCS\_LENGTH耗时Ο(mn)。



3）源码 + 注释

具体源码见附件。

4）算法测试结果

经测试，算法能正确求出最长公共子序列：

