



## 6.7项目实战

# 设计实验报告防抄袭系统

需求：

同学们的实验报告抄袭现象严重，现为了防止实验报告抄袭的恶习，让真正撰写实验报告的组能够获得公平的分数，需要设计一个系统能够查找两个实验报告中相同的文字内容，从而计算两个实验报告的相似度。

问题分析：怎样计算两个实验报告的相似度？

$$S = \frac{\text{相同的字数}}{\text{总字数}}$$

$$S = \frac{\text{相同的模块的字数}}{\text{总字数}}$$

# 设计实验报告防抄袭系统

该问题以公司给推销员的各位顾客的推销难度评分和推销员的位次作为输入，以合适的要推销的目标作为输出，要求查询速度尽量快的找到推销员要推销的目标进行推销。从算法的角度看，实际上就是让我们对各位顾客进行以推销难度评分从低到高排序，然后选择合适第k位推销员的顾客。

文档1

问题以公司给推销员的各位顾客的推销难度评分和推销员的位次作为输入。合适的要推销的目标作为输出。实际上就是让我们对各位顾客进行以推销难度评分从低到高排序，然后选择合适第k位推销员的顾客。

文档2

# 设计实验报告防抄袭系统

- 假设我们将每个模块看成一个整体，以一个字母来代表，不考虑文字顺序变换这一抄袭手法，只考虑文字的增添和减少，则问题是否可以简化为以下问题？
- 给定2个字符序列X和Y，当另一序列Z既是X的子序列又是Y的子序列时，称Z是序列X和Y的**公共子序列**。找到两个序列的最长公共子序列，其长度也就是两个序列中最长相同的文字模块数目

# 设计实验报告防抄袭系统

- 假设我们将每个模块看成一个整体，以一个方案去代替，不考虑方案顺序变换这一抄问，考虑其变化。减少，则
- 
- 给序列X和Y，当Z既是X的子序列时，称Z是序列X和Y的公共子序列。找到两个序列的最长公共子序列，其长度也就是两个序列中最长相同的文字模块数目

# 最长公共子序列

- 若给定序列 $X=\{x_1, x_2, \dots, x_m\}$ ，则另一序列 $Z=\{z_1, z_2, \dots, z_k\}$ ，是 $X$ 的子序列是指存在一个严格递增下标序列 $\{i_1, i_2, \dots, i_k\}$ 使得对于所有 $j=1, 2, \dots, k$ 有： $z_j=x_{i_j}$ 。
- 例如，序列 $Z=\{B, C, D, B\}$ 是序列 $X=\{A, B, C, B, D, A, B\}$ 的子序列，相应的递增下标序列为 $\{2, 3, 5, 7\}$ 。
- 给定2个序列 $X$ 和 $Y$ ，当另一序列 $Z$ 既是 $X$ 的子序列又是 $Y$ 的子序列时，称 $Z$ 是序列 $X$ 和 $Y$ 的**公共子序列**。  
例如： $X=\{A, B, C, B, D, A, B\}$ ,  $Y=\{B, D, C, A, B, A\}$   
     $\{B, C, A\}$   
     $\{B, C, B, A\}$

# 最长公共子序列的结构

**问题：** 给定2个序列 $X=\{x_1, x_2, \dots, x_m\}$ 和 $Y=\{y_1, y_2, \dots, y_n\}$ ，找出X和Y的**最长公共子序列**。

**怎样寻找子问题？目标：** 具有最优子结构的子问题划分



# 最长公共子序列的结构

**问题：**给定2个序列 $X=\{x_1, x_2, \dots, x_m\}$ 和 $Y=\{y_1, y_2, \dots, y_n\}$ ，找出X和Y的**最长公共子序列**。

**怎样寻找子问题？目标：具有最优子结构的子问题划分**

设序列 $X=\{x_1, x_2, \dots, x_m\}$ 和 $Y=\{y_1, y_2, \dots, y_n\}$ 的最长公共子序列为 $Z=\{z_1, z_2, \dots, z_k\}$ ，则

- (1)若 $x_m = y_n$ ，则 $z_k = x_m = y_n$ ，且 $z_{k-1}$ 是 $X_{m-1}$ 和 $Y_{n-1}$ 的**最长公共子序列**。
- (2)若 $x_m \neq y_n$ 且 $z_k \neq x_m$ ，则Z是 $X_{m-1}$ 和Y的**最长公共子序列**。
- (3)若 $x_m \neq y_n$ 且 $z_k \neq y_n$ ，则Z是X和 $Y_{n-1}$ 的**最长公共子序列**。



# 最长公共子序列的结构

证明:

(1) 用反证法。若 $z_k \neq x_m$ , 则 $\{z_1, z_2, \dots, z_k, x_m\}$ 是 $X$ 和 $Y$ 的长度为 $k+1$ 的公共子序列。这与 $Z$ 是 $X$ 和 $Y$ 的最长公共子序列矛盾。因此, 必有 $z_k = x_m = y_n$ 。由此可知 $Z_{k-1}$ 是 $X_{m-1}$ 和 $Y_{n-1}$ 的长度为 $k-1$ 的公共子序列。若 $X_{m-1}$ 和 $Y_{n-1}$ 有长度大于 $k-1$ 的公共子序列 $W$ , 则将 $x_m$ 加在其尾部产生 $X$ 和 $Y$ 的长度大于 $k$ 的公共子序列, 与此矛盾。故 $Z_{k-1}$ 是 $X_{m-1}$ 和 $Y_{n-1}$ 的最长公共子序列。

(2) 由于 $z_k \neq x_m$ ,  $Z$ 是 $X_{m-1}$ 和 $Y$ 的公共子序列。若 $X_{m-1}$ 和 $Y$ 有长度大于 $k$ 的公共子序列 $W$ , 则 $W$ 也是 $X$ 和 $Y$ 的长度大于 $k$ 的公共子序列。这与 $Z$ 是 $X$ 和 $Y$ 的最长公共子序列矛盾。故 $Z$ 是 $X_{m-1}$ 和 $Y$ 的最长公共子序列。

由此可见, 2个序列的最长公共子序列包含了这2个序列的前缀的最长公共子序列。因此, 最长公共子序列问题具有**最优子结构性质**。

# 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性质建立子问题**最优值**的递归关系。用 $c[i][j]$ 记录序列 $X$ 和 $Y$ 的最长公共子序列的长度。其中,  $X_i = \{x_1, x_2, \dots, x_i\}$ ;  $Y_j = \{y_1, y_2, \dots, y_j\}$ 。当 $i=0$ 或 $j=0$ 时, 空序列是 $X_i$ 和 $Y_j$ 的最长公共子序列。故此时 $C[i][j]=0$ 。其他情况下, 由最优子结构性性质可建立递归关系如下:

$$c[i][j] = \begin{cases} 0 & i = 0, j = 0 \\ c[i-1][j-1] + 1 & i, j > 0; x_i = y_j \\ \max\{c[i][j-1], c[i-1][j]\} & i, j > 0; x_i \neq y_j \end{cases}$$

情况1

重叠子问题:  $X_{m-1}$ 和 $Y_{m-1}$ 最长公共子序列

情况  
2,3

# 计算最优值

由于在所考虑的子问题空间中，总共有 $\theta(mn)$ 个不同的子问题，因此，用动态规划算法自底向上地计算最优值能提高算法的效率。

## Algorithm lcsLength(x,y,b)

```
1: m ← x.length-1;
2: n ← y.length-1;
3: c[i][0]=0; c[0][i]=0;
4: for (int i = 1; i <= m; i++)
5:   for (int j = 1; j <= n; j++)
6:     if (x[i]==y[j])
7:       c[i][j]=c[i-1][j-1]+1;
8:       b[i][j]=1;
9:     else if (c[i-1][j]>=c[i][j-1])
10:      c[i][j]=c[i-1][j];
11:      b[i][j]=2;
12:     else
13:      c[i][j]=c[i][j-1];
14:      b[i][j]=3;
```

## 构造最长公共子序列

## Algorithm lcs(int i,int j,char [] x,int [][] b)

```
{
  if (i ==0 || j==0) return;
  if (b[i][j]== 1){
    lcs(i-1,j-1,x,b);
    System.out.print(x[i]);
  }
  else if (b[i][j]== 2) lcs(i-1,j,x,b);
  else lcs(i,j-1,x,b);
}
```

# 思考?

- 该问题以，以合适的要推销的目标作为输出，公司给推销员的各位顾客的推销难度评分和推销员的位次作为输入要求查询速度尽量快的找到推销员要推销的目标进行推销。从算法的角度看，实际上就是让我们对各位顾客进行以推销难度评分从低到高排序，然后选择合适第 $k$ 位推销员的顾客。

文档1

问题以公司给推销员的各位顾客的推销难度评分和推销员的位次作为输入。合适的要推销的目标作为输出。实际上就是让我们对各位顾客进行以推销难度评分从低到高排序，然后选择合适第 $k$ 位推销员的顾客。

文档2

# 思考？

针对文字顺序变换这一抄袭现象，  
如何改进算法，或进行设计，实  
现文字顺序变换的检测？