练习5 – 最长公共子串

- 最长公共子串（Longest Common Substring）
  - 子串是连续的子序列

- 求两个字符串的最长公共子串长度
  - ABCBA 和 BABCA 的最长公共子串是 ABC，长度为 3

最长公共子串 – 思路

- 假设 2 个字符串分别是 str1、str2
  - i ∈ [1, str1.length]
  - j ∈ [1, str2.length]

- 假设 dp(i, j) 是以 str1[i – 1]、str2[j – 1] 结尾的最长公共子串长度
  - dp(i, 0)、dp(0, j) 初始值均为 0
  - 如果 str1[i – 1] = str2[j – 1]，那么 dp(i, j) = dp(i – 1, j – 1) + 1
  - 如果 str1[i – 1] ≠ str2[j – 1]，那么 dp(i, j) = 0

- 最长公共子串的长度是所有 dp(i, j) 中的最大值 max { dp(i, j) }

小码哥教育 SEEMYGO

最长公共子串 – 实现

```
int lcs(String str1, String str2) {
    if (str1 == null || str2 == null) return 0;
    char[] cs1 = str1.toCharArray();
    if (cs1.length == 0) return 0;
    char[] cs2 = str2.toCharArray();
    if (cs2.length == 0) return 0;
    int[][] dp = new int[cs1.length + 1][cs2.length + 1];
    int max = 0;
    for (int i = 1; i <= cs1.length; i++) {
        for (int j = 1; j <= cs2.length; j++) {
            if (cs1[i - 1] != cs2[j - 1]) continue;
            dp[i][j] = dp[i - 1][j - 1] + 1;
            max = Math.max(max, dp[i][j]);
        }
    }
    return max;
}
```

- 空间复杂度：$O(n * m)$
- 时间复杂度：$O(n * m)$

最长公共子串 – 实现

- dp 数组的计算结果如下所示

			B	A	B	C	A
	i \ j	0	1	2	3	4	5
	0	0	0	0	0	0	0
A	1	0	0	1	0	0	1
B	2	0	1	0	2	0	0
C	3	0	0	0	0	3	0
B	4	0	1	0	1	0	0
A	5	0	0	2	0	0	1

最长公共子串 – 一维数组实现

```
if (str1 == null || str2 == null) return 0;
char[] cs1 = str1.toCharArray();
if (cs1.length == 0) return 0;
char[] cs2 = str2.toCharArray();
if (cs2.length == 0) return 0;
char[] rowStr = cs1, colStr = cs2;
if (cs1.length < cs2.length) {
    colStr = cs1;
    rowStr = cs2;
}
int[] dp = new int[colStr.length + 1];
```

```
int max = 0;
for (int i = 1; i <= rowStr.length; i++) {
    for (int j = colStr.length; j >= 1; j--) {
        if (rowStr[i - 1] == colStr[j - 1]) {
            dp[j] = dp[j - 1] + 1;
        } else {
            dp[j] = 0;
        }
        max = Math.max(max, dp[j]);
    }
}
return max;
```

- 空间复杂度：$O(k),\ k = \min\{n, m\}$
- 时间复杂度：$O(n * m)$