

## 79. Longest Common Substring

Given two strings, find the longest common substring. Return the length of it.

思维：两层循环，用二维数组记录(动态规划)

列	a	b	c	d
b	0	1	0	0
c	0	0	2	0
a	1	0	0	0

```
1 public int longestCommonSubstring(String A, String B) {
2     if( null == A || null == B || A.length() == 0 || B.length() == 0 ) return
    0;
3     int[][] D = new int[A.length()][B.length()];
4     int max = 0;
5     for( int i = 0; i < A.length(); i++)
6         for( int j = 0; j < B.length(); j++ ){
7             if( A.charAt(i) == B.charAt(j) ){
8                 if( 0 == i || 0 == j )
9                     D[i][j] = 1;
10                else
11                    D[i][j] = D[i-1][j-1] + 1;
12                max = Math.max( max, D[i][j] );
13            } else
14                D[i][j] = 0;
15        }
16    return max;
17 }
```

思维二：在二维数组中，当前行的数据只依赖于上一行，所以可以只用一维数组来替代。注意，一维数组后面的数是依赖前面的数，所以在更新数组时，需要从后往前更新。

```
1     for( int i = 0; i < A.length(); i++)
2         for( int j = B.length() - 1; j >=0; j-- ){
3             if( A.charAt(i) == B.charAt(j) ){
4                 if( 0 == i || 0 == j )
5                     D[j] = 1;
6                 else
7                     D[j] = D[j-1] + 1;
8                 max = Math.max( max, D[j] );
9             } else
10                D[j] = 0;
11        }
```