**动态规划：**

|  |
| --- |
| 考虑当前nums[i]时，最长增长子序列是多少，dp[i] 存放的是取第i个位置时的最长序列  状态转移方程：从nums[i]位置的左边数去找比nums[i] 小的数，找到就更新为 dp[i] = max(dp[i],dp[j] + 1)      与上一题思路相同      使用两个dp数组,一个记录正摆动，一个记录负摆动，当考虑当前nums[i] 时，如果nums[i] - nums[i - 1] >0  则更新正摆动P[i] = N[i] + 1,如果小于，则更新负摆动N[i] = P[i] + 1;    优化：      将text1和text2的字符串转变成以text1各个字符为行，text2各个字符为列的二维矩阵，dp[i][j]记录text1.charAt(i)是否等于 text2.charAt(j)时相应更新他的最大值 |