```
class Solution {
    public int longestPalindromeSubseq(String s) {
        int[][] dp = new int[s.length()][s.length()];
        for (int i = s.length() - 1; i >= 0; i--) {
            dp[i][i] = 1;
            for (int j = i+1; j < s.length(); j++) {
                if (s.charAt(i) == s.charAt(j)) {
                    dp[i][j] = 2 + dp[i+1][j-1];
                } else {
                    dp[i][j] = Math.max(dp[i+1][j], dp[i][j-1]);
                }
            }
        }
        return dp[0][s.length()-1];
    }
}
```

Given a string s, find the longest palindromic subsequence's length in s. You may assume that the maximum length of s is 1000.

Example 1:
Input:

```
"bbbab"
```

Output:
`4`

One possible longest palindromic subsequence is "bbbb".