```
dp[i, j] means s.substring(i, j+1) is a palin string
it is s.charAt(i) == s.charAt(j) && s.substring(i+1, j)
```

```
class Solution {
    public String longestPalindrome(String s) {
        boolean[][] dp = new boolean[s.length()]
[s.length()];
        String res = "";

        for (int i = s.length()-1; i >= 0; i--) {
            for (int j = i; j < s.length(); j++) {
                if (s.charAt(i) == s.charAt(j) && (j - i <
3 || dp[i+1][j-1]))
                    dp[i][j] = true;
                if (dp[i][j] && res.length() < j - i + 1) {
                    res = s.substring(i, j+1);
                }
            }
        }
        return res;
    }
}
```

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

```
Input: "babad"
Output: "bab"
Note: "aba" is also a valid answer.
```

Example 2:

```
Input: "cbbd"
Output: "bb"
```