

LeetCode647 回文子串

给定一个字符串，你的任务是计算这个字符串中有多少个回文子串。
具有不同开始位置或结束位置的子串，即使是由相同的字符组成，也会被计为是不同的子串。

$dp[i][j]$ 表示字符串从 i 到 j 是不是回文子串

$$dp[i][j] = dp[i+1][j-1] \& true$$

- i 需要从大到小
- j 需要从小到大
- 初始化

```
int countSubstrings(string s) {
    int n = s.length();
    vector<vector<bool>> dp(n, vector<bool>(n, false));

    // 初始化
    for (int i = 0; i < n; i++) {
        dp[i][i] = true;
    }
    for (int i = 0; i < n - 1; i++) {
        if (s[i] == s[i + 1]) {
            dp[i][i + 1] = true;
        }
    }

    // 状态转移
    for (int j = 0; j < n; j++) {
        for (int i = 0; i < j - 1; i++) {
            if (s[i] == s[j]) {
                dp[i][j] = dp[i + 1][j - 1] & true;
            }
            else {
                dp[i][j] = false;
            }
        }
    }

    // 计算结果
    int res = 0;
    for (int i = 0; i < n; i++) {
```

```
        for (int j = 0; j < n; j++) {  
            if (dp[i][j] && i <= j) {  
                res++;  
            }  
        }  
    }  
    return res;  
}
```