



```

1  class Solution {
2  public:
3      bool isMatch(string s, string p) {
4          int m = s.size();
5          int n = p.size();
6
7          s = " " + s;
8          p = " " + p;
9          vector<vector<bool>> dp(m + 1, vector<bool>(n + 1, false));
10         dp[0][0] = true;
11
12         for (int i = 0; i <= m; i++) {
13             for (int j = 1; j <= n; j++) {
14                 if (p[j] == '.') {
15                     if (i > 0)
16                         dp[i][j] = dp[i - 1][j - 1];
17                 } else if (p[j] == '*') {
18                     dp[i][j] = dp[i][j - 2]; // * 匹配 0 个前一字符
19                     if (i > 0 && (s[i] == p[j - 1] || p[j - 1] == '.')) {
20                         dp[i][j] = dp[i][j] ||
21                             dp[i - 1][j]; // * 匹配 1 个或多个前一字符
22                     }
23                 } else {
24                     if (i > 0 && s[i] == p[j]) {
25                         dp[i][j] = dp[i - 1][j - 1];
26                     }
27                 }
28             }
29         }
30
31         // for (int i = 1; i <= m; i++) {
32         //     for (int j = 1; j <= n; j++) {
33         //         if (s[i] == p[j] && dp[i - 1][j - 1])
34         //             dp[i][j] = true;
35         //         else if (p[j] == '.')
36         //             dp[i][j] = dp[i - 1][j - 1];
37         //         else if (p[j] == '*') {
38         //             dp[i][j] = dp[i][j - 2]; // * 匹配 0 个前一字符
39         //             if (i > 0 && (s[i] == p[j - 1] || p[j - 1] == '.'))

```

```
40         //      {
41         //          dp[i][j] = dp[i][j] || dp[i - 1][j]; // * 匹配 1
42         //          个或多个前一字符
43         //      }
44         //      // if(p[j-1] == '.')
45         //      // dp[i][j] = dp[i][j-2] || dp[i-1][j];
46         //      // else
47         //      // dp[i][j] = dp[i][j-2] || (s[i] = p[j-1] &&
48         //      // dp[i-1][j]);
49         //      }
50     // }
51 // }
52 return dp[m][n];
53 }
54 };
```