# 19. 正则表达式匹配

[牛客网](https://www.nowcoder.com/practice/28970c15befb4ff3a264189087b99ad4?tpId=13&tqId=11205&tab=answerKey&from=cyc_github)

## 题目描述

请实现一个函数用来匹配包括 '.' 和 '\*' 的正则表达式。模式中的字符 '.' 表示任意一个字符，而 '\*' 表示它前面的字符可以出现任意次（包含 0 次）。

在本题中，匹配是指字符串的所有字符匹配整个模式。例如，字符串 "aaa" 与模式 "a.a" 和 "ab\*ac\*a" 匹配，但是与 "aa.a" 和 "ab\*a" 均不匹配。

## 解题思路

应该注意到，'.' 是用来当做一个任意字符，而 '\*' 是用来重复前面的字符。这两个的作用不同，不能把 '.' 的作用和 '\*' 进行类比，从而把它当成重复前面字符一次。

public boolean match(String str, String pattern) {

 int m = str.length(), n = pattern.length();
 boolean[][] dp = new boolean[m + 1][n + 1];

 dp[0][0] = true;
 for (int i = 1; i <= n; i++)
 if (pattern.charAt(i - 1) == '\*')
 dp[0][i] = dp[0][i - 2];

 for (int i = 1; i <= m; i++)
 for (int j = 1; j <= n; j++)
 if (str.charAt(i - 1) == pattern.charAt(j - 1) || pattern.charAt(j - 1) == '.')
 dp[i][j] = dp[i - 1][j - 1];
 else if (pattern.charAt(j - 1) == '\*')
 if (pattern.charAt(j - 2) == str.charAt(i - 1) || pattern.charAt(j - 2) == '.') {
 dp[i][j] |= dp[i][j - 1]; // a\* counts as single a
 dp[i][j] |= dp[i - 1][j]; // a\* counts as multiple a
 dp[i][j] |= dp[i][j - 2]; // a\* counts as empty
 } else
 dp[i][j] = dp[i][j - 2]; // a\* only counts as empty

 return dp[m][n];
}