Question: https://leetcode.com/problems/regular-expression-matching/submissions/

1, If p.charAt(j) == s.charAt(i) : dp[i][j] = dp[i-1][j-1];

2, If p.charAt(j) == '.' : dp[i][j] = dp[i-1][j-1];

3, If p.charAt(j) == '\*':

here are two sub conditions:

1 if p.charAt(j-1) != s.charAt(i) : dp[i][j] = dp[i][j-2] //in this case, a\* only counts as empty

2 if p.charAt(i-1) == s.charAt(i) or p.charAt(i-1) == '.':

dp[i][j] = dp[i-1][j] //in this case, a\* counts as multiple a

or dp[i][j] = dp[i][j-1] // in this case, a\* counts as single a

or dp[i][j] = dp[i][j-2] // in this case, a\* counts as empty

Code:  
class Solution {

public boolean isMatch(String s, String p) {

if (s == null || p == null) {

return false;

}

boolean[][] dp = new boolean[s.length()+1][p.length()+1];

dp[0][0] = true;

for (int i = 0; i < p.length(); i++) {

if (p.charAt(i) == '\*' && dp[0][i-1]) {

dp[0][i+1] = true;

}

}

for (int i = 0 ; i < s.length(); i++) {

for (int j = 0; j < p.length(); j++) {

if (p.charAt(j) == '.') {

dp[i+1][j+1] = dp[i][j];

}

if (p.charAt(j) == s.charAt(i)) {

dp[i+1][j+1] = dp[i][j];

}

if (p.charAt(j) == '\*') {

if (p.charAt(j-1) != s.charAt(i) && p.charAt(j-1) != '.') {

dp[i+1][j+1] = dp[i+1][j-1];

} else {

dp[i+1][j+1] = (dp[i+1][j] || dp[i][j+1] || dp[i+1][j-1]);

}

}

}

}

return dp[s.length()][p.length()];

}

}

Github Link :<https://lnkd.in/ecwtJeaz>