[**Regular Expression Matching**](https://leetcode.com/problems/regular-expression-matching/)

**public** **class** RegularExpressionMatching {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(*isMatch*("aa", "a"));

}

**public** **static** **boolean** isMatch(String s, String p) {

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

**return** **true**;

}

**int** s\_length = s.length();

**int** p\_length = p.length();

**boolean**[][] dp = **new** **boolean**[s\_length + 1][p\_length + 1];

dp[0][0] = **true**;

**for**(**int** i = 2 ; i <= p\_length ; i++) {

**if**(p.charAt(i-1) == '\*') {

dp[0][i] = dp[0][i - 2];

}

}

**for**(**int** i = 1 ; i <= s\_length ; i++) {

**for**(**int** j = 1 ; j <= p\_length ; j++) {

**if**(s.charAt(i - 1) == p.charAt(j - 1) || p.charAt(j - 1) == '.') {

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

}

**else** **if**(p.charAt(j - 1) == '\*') {

dp[i][j] = dp[i][j - 2];

**if**(s.charAt(i - 1) == p.charAt(j - 2) || p.charAt(j - 2) == '.') {

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

}

}

**else** {

dp[i][j] = **false**;

}

}

}

**return** dp[s\_length][p\_length];

}

}

Time Complexity : O(n \* m) where n is length of string s and m is length of pattern p

Space Complexity : O(n \* m) where n is length of string s and m is length of pattern p