# 10. Regular Expression Matching

Given an input string s and a pattern p, implement regular expression matching with support for '.' and '\*' where:

'.' Matches any single character.​​​​

'\*' Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

## SOLUTION IN C++

class Solution {

public boolean isMatch(String s, String p) {

final int m = s.length();

final int n = p.length();

boolean[][] dp = new boolean[m + 1][n + 1];

dp[0][0] = true;

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

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

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

for (int i = 0; i < m; ++i)

for (int j = 0; j < n; ++j)

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

final boolean noRepeat = dp[i + 1][j - 1];

final boolean doRepeat = isMatch(s, i, p, j - 1) && dp[i][j + 1];

dp[i + 1][j + 1] = noRepeat || doRepeat;

} else if (isMatch(s, i, p, j)) {

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

}

return dp[m][n];

}

private boolean isMatch(final String s, int i, final String p, int j) {

return j >= 0 && p.charAt(j) == '.' || s.charAt(i) == p.charAt(j);

}

}