**Regular expression matching**

Implement regular expression matching with support for '.' and '\*'.

class Solution **{**

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

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

mat**[**0**][**0**]** **=** **true; //The other are automatically false.**

**for(**int i**=**0**;**i**<**p**.**length**();**i**++){ //Making sure corner cases pass.**

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

mat**[**0**][**i**+**1**]=true;**

**}**

**}**

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

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

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

mat**[**i**+**1**][**j**+**1**]** **=** mat**[**i**][**j**];**

**}**

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

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

mat**[**i**+**1**][**j**+**1**]=**mat**[**i**+**1**][**j**-**1**];**

**}else{**

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

**}**

**}**

**}**

**}**

**return** mat**[**s**.**length**()][**p**.**length**()];**

**}**

**}**

* In this problem I used Dynamic Programming.