题目：

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

'.' Matches any single character.

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

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

The function prototype should be:

bool isMatch(const char \*s, const char \*p)

Some examples:

isMatch("aa","a") → false

isMatch("aa","aa") → true

isMatch("aaa","aa") → false

isMatch("aa", "a\*") → true

isMatch("aa", ".\*") → true

isMatch("ab", ".\*") → true

isMatch("aab", "c\*a\*b") → true

题目的意思就是模拟正则表达式“.\*”的匹配过程

我们首先要知道“.\*”的匹配过程，“.“代表的是任意字符，“\*”代表0个或者多个前面的字符，例如：a\*b代表的是：b（0个a）、ab、aab 、 … 、 aa~b

需要区分的是a+b和a\*b的区别

“+“代表的是一个或者多个前面的字符。

例如a+b代表的是：ab、aab、… 、aa~b

思路：

采用递归方法

具体代码如下：

public boolean isMatch(String text, String pattern) {

if (pattern.isEmpty())

return text.isEmpty();

boolean first\_match =

(!text.isEmpty() && (pattern.charAt(0) == text.charAt(0) || pattern.charAt(0) == '.'));

if (pattern.length() >= 2 && pattern.charAt(1) == '\*'){

return (isMatch(text, pattern.substring(2)) ||

(first\_match && isMatch(text.substring(1), pattern)));

} else {

return first\_match && isMatch(text.substring(1), pattern.substring(1));

}

}

//标红线的代码看不懂啊，不知道系统怎么执行的。具体思路就不写了，以后有时间回来看再进行编写。