

# 暨南大学本科实验报告专用纸(附页)

---

## 实现一个高性能的基础正则表达式匹配方法

\* 实验项目类型：设计性

\*此表由学生按顺序填写

课程名称 面向对象程序设计/JAVA 语言 成绩评定           

实验项目名称 实现一个高性能的基础正则表达式匹配方法

指导老师 干晓聪

实验项目编号 1 实验项目类型 设计性 实验地点 数学系机房

学生姓名 郭彦培 学号 2022101149

学院 信息科学技术学院 系 数学系 专业 信息管理与信息系统

实验时间 2023 年 11 月 1 日上午 ~ 2023 年 11 月 1 日中午

### 一、实验目的

练习字符串相关操作与基础数组的使用

### 二、实验环境

计算机：PC X64

操作系统：Windows

编程语言：Java

IDE：Visual Studio Code

在线测试平台：leetcode

### 三、程序原理

实现的正则规则为 `.` 与 `*`

`.` 匹配任意单个字符, `*` 匹配零个或多个前面的那一个元素

算法使用动态规划, 复杂度为严格线性, 即  $O(N)$

具体实现的过程使用了 `String` 类、`boolean` 数组

# 暨南大学本科实验报告专用纸(附页)

---

代码正确性已通过 leetcode 平台测试

## 四、程序代码

文件 `sis4\regularExp` 实现了一个 `boolean` 函数，用于字符串匹配

```
package sis4;

public class regularExp {

    public static boolean isMatch(String s, String p) {
        int lenS = s.length();
        int lenP = p.length();
        boolean[][] m = new boolean[lenS+1][lenP+1];
        m[0][0] = true;
        for(int i = 1; i <= lenS; i++){
            m[i][0] = false;
        }
        for(int j = 1; j <= lenP; j++){
            m[0][j] = false;
            if(j>=2 && p.charAt(j-1)=='*'){
                m[0][j] = m[0][j-2];
            }
        }
        for(int i = 1; i <= lenS; i++){
            for(int j = 1; j <= lenP; j++){
                if(s.charAt(i-1) == p.charAt(j-1) || p.charAt(j-1) == '.'){
                    m[i][j] = m[i-1][j-1];
                }
                else if(p.charAt(j-1) == '*'){
                    if(p.charAt(j-2) == s.charAt(i-1) || p.charAt(j-2) == '.'){
                        m[i][j] = m[i-1][j] || m[i][j-2];
                    }
                    else{
                        m[i][j] = m[i][j-2];
                    }
                }
            }
        }
        return m[lenS][lenP];
    }
}
```

## 五、 出现的问题、原因与解决方法

一开始二维数组 `m` 使用 `ArrayList` 实现，随后在后续的编码过程中发现操作过于复杂。

例如对于 `boolean` 数组，`m[0][j] = m[i-1][j] || m[i][j-2]` 这段代码，若使用 `ArrayList` 则需要写成

```
m.get(i).set(j, Boolean.valueOf(((boolean)m.get(i-1).get(j) || (boolean)m.get(i).get(j-2))));
```

研究过后发现，在不太需要访问控制的场合，适当使用数组而非对象会提高编码效率。

于是本段代码没有使用 `ArrayList`。

## 六、 测试数据与运行结果

输入	输出	解释
<code>s = "ab", p = ".*"</code>	<code>true</code>	“ <code>.*</code> ”表示可匹配零个或多个（ <code>*</code> ）任意字符（ <code>.</code> ）。
<code>s = "aa", p = "a"</code>	<code>false</code>	“ <code>a</code> ”无法匹配“ <code>aa</code> ”整个字符串。
<code>s = "accomplish", p = "ac*m.l.*"</code>	<code>true</code>	<code>*</code> 匹配为上一个 <code>c</code> ， <code>.</code> 匹配为 <code>m</code> 之后 <code>.*</code> 匹配为 <code>ish</code>

注：测试平台 `leetcode` 的特性为直接向函数传参，因此不需要实现输入输出。