

暨南大学本科实验报告专用纸(附页)

实现一个 trie(字典树)模板

* 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 trie(字典树)模板 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 10 月 27 日上午 ~ 2023 年 10 月 27 日中午

一、实验目的

练习字符串相关操作

二、实验环境

计算机：PC X64

操作系统：Windows

编程语言：Java

IDE：Visual Studio Code

三、程序原理

字典树，英文名 Trie，顾名思义为类似字典的树状结构。

字典树的每条边代表一个字母，从根节点到树上某个节点的路径则代表了一个字符串。我们用 $\theta(u, c)$ 表示节点 u 的 c 字符指向的下一个节点，或者说节点 u 代表的字符串后添加一个字符 c 形成的字符串的节点。

字典树最基础的应用为：查找一个字符串是否在字典中出现过，即 \exists 字符串 s 和一个索引 i ， $s.t. \forall c < s.size(), \text{tire}(s + c) = s(c)$

暨南大学本科实验报告专用纸(附页)

具体实现的过程使用了 `String` 类、`ArrayList` 类、`HashMap` 类

四、程序代码

文件 `sis3\TrieTree` 实现了字典树类

```
package sis3;

import java.util.ArrayList;
import java.util.HashMap;

public class TrieTree {
    public ArrayList<HashMap<Character, Integer>> t;
    public int root = 0;

    public TrieTree()
    {
        t = new ArrayList<HashMap<Character, Integer>>();
        t.add(new HashMap<Character, Integer>());
    }

    public void addedge(String _s)
    {
        int pvidx = root;
        _s += '-';
        for (int i = 0; i < _s.length(); i++) {
            if (t.get(pvidx).containsKey(_s.charAt(i))) {
                pvidx = t.get(pvidx).get(_s.charAt(i));
            } else {
                t.get(pvidx).put(_s.charAt(i), t.size());
                t.add(new HashMap<Character, Integer>());
                pvidx = t.get(pvidx).get(_s.charAt(i));
            }
        }
    }

    public boolean ifcmp(String s)
    {
        int pvidx = root;
        for (int i = 0; i < s.length(); i++) {
            if (t.get(pvidx).containsKey(s.charAt(i)))
                pvidx = t.get(pvidx).get(s.charAt(i));
            else
                return false;
        }
    }
}
```

暨南大学本科实验报告专用纸(附页)

```
    }  
    return t.get(pvidx).containsKey('-');  
  }  
}
```

文件 `sis\Text.java` 实现了输入输出与测试数据处理

```
package sis3;  
  
import java.util.Scanner;  
  
// 输入一行一个整数 T, 表示数据组数 T 对于每组数据, 格式如下  
// 第一行是两个整数, 分别表示模式串的个数 n 和询问的个数 q  
// 接下来 n 行, 每行一个模式串, 表示模式串集合 接下来 q 行, 每行一个询问, 表示询问集合  
public class Test {  
    public static void main(String[] args) {  
        try (Scanner sc = new Scanner(System.in)) {  
            int T = sc.nextInt();  
            TrieTree t = new TrieTree();  
            while (T --> 0) {  
                int n = sc.nextInt();  
                int q = sc.nextInt();  
                sc.nextLine();  
                while (n-- > 0) {  
                    String s = sc.nextLine();  
                    t.addedge(s);  
                }  
                while (q-- > 0) {  
                    String s = sc.nextLine();  
                    if (t.ifcmp(s))  
                        System.out.println("YES");  
                    else  
                        System.out.println("NO");  
                }  
            }  
        }  
    }  
}
```

五、 出现的问题、原因与解决方法

编码过程十分顺利

六、 测试数据与运行结果

输入	输出	解释
3		输入测试样例数量
3 3 fusufusu fusu anguei		记录测试用字典与测试字符串数量
fusu	2	字典中有两个字符串满足前缀为 fusu
anguei	1	字典中有一个字符串满足前缀为 anguei
kkksc	0	字典中有零个字符串满足前缀为 kkksc
5 2 fusu Fusu AFakeFusu afakefusu fusuisnotfake		记录第二个测试样例 测试用字典与测试字符串数量
Fusu	1	字典中有一个字符串满足前缀为 Fusu
fusu	2	字典中有两个字符串满足前缀为 fusu
1 1 998244353		记录第三个测试样例 测试用字典与测试字符串数量
9	1	字典中有一个字符串满足前缀为 9