

# 暨南大学本科实验报告专用纸(附页)

## 字典树 Trie

课程名称 数据结构 成绩评定   
实验项目名称 字典树 Trie 指导老师 干晓聪  
实验项目编号 12 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 2024年6月13日上午 ~ 2024年7月13日中午

## 1. 实验目的

实现一个字典树

## 2. 实验环境

计算机: PC X64

操作系统: Windows + Ubuntu20.0LTS

编程语言: C++: GCC std20

IDE: Visual Studio Code

## 3. 程序原理

基于一棵每层至多 26 分支的搜索树, 实现在已有字典的基础上, 查询字符串  $s$  在字典中是否出现。

规定每个节点  $t$  一一对应一个字母  $c_t$ ,  $t$  的子节点列表  $S_t$

则  $\exists t' \in S_t \Leftrightarrow$  在字典中有  $c_t$  的后缀为  $c_{t'}$

因此如果存在路径  $t_0 \rightarrow t_1 \dots t_{|s|} s.t. \forall i \in [1, |s|]$  有  $t_i = s[i]$ , 则  $s$  在字典中出现过。

容易证明, 查询复杂度为  $\mathcal{O}(|s|)$ , 与字典大小无关。

对于字典规模  $n$ , 建立时的时间复杂度为  $\mathcal{O}(n)$ , 字典树空间复杂度为  $\mathcal{O}(n \log_{26} n)$

## 4. 程序代码

### 4.1. template\_overAll.h

```
1  #include <vector>
2  #include <map>
3  #include <string>
4  #include <string.h>
5  #include <math.h>
6  #include <set>
7  #include <algorithm>
8  #include <iostream>
9  #include <queue>
10
11 using namespace std;
12
13 #define ll long long
14 #define pb push_back
15 #define ld long double
16 const ll int maxn = 1E5+10;
17 const ll int mod1 = 998244353;
18 const ll int mod2 = 1E9+7;
19
20 #define _IN_TEMPLATE_
21
22 ll int str2int(string s)
23 {
24     ll int rec = 0;
25     ll int pw = 1;
26     for(int i = s.length()-1; i >= 0; i --)
27     {
28         int gt = s[i] - '0';
29         if(gt < 0 || gt > 9) return INT64_MAX;
30         rec += gt * pw;
31         pw *= 10;
32     }
33     return rec;
34 }
35
36 vector<ll int> testReadLine()
37 {
38     string s;
39     getline(cin,s);
40     s.push_back(' ');
41     vector<ll int> rearr;
42     vector<string> substring;
43     string ts;
44     for(int i = 0; i < s.size(); i ++){
45         if(s[i] == ' '){
46             substring.push_back(ts);
```

# 暨南大学本科实验报告专用纸(附页)

---

```
48         ts.clear();
49     } else ts.push_back(s[i]);
50 }
51 for(int i = 0; i < substring.size(); i +
52 +)rearr.push_back(str2int(substring[i]));
53 return rearr;
54 }
```

## 4.2. trie\_Tree.h

```
1  #include <template_overAll.h>
2
3  class Trie//AC
4  {
5  public:
6      vector<map<char, int>> t;
7      int root = 0;
8      Trie()
9      {
10         t.resize(1);
11     }
12     void addedge(string _s)
13     {
14         int pidx = root;
15         _s.push_back('-');
16         for (int i = 0; i < _s.size(); i++)
17         {
18             if (t[pidx].find(_s[i]) != t[pidx].end())
19             {
20                 pidx = t[pidx][_s[i]];
21             }
22             else
23             {
24                 t[pidx][_s[i]] = t.size();
25                 t.push_back(map<char, int>());
26                 pidx = t[pidx][_s[i]];
27             }
28         }
29     }
30     bool ifcmp(string &s)
31     {
32         int pidx = root;
33         for(int i = 0; i < s.size(); i++)
34         {
35             if(t[pidx].find(s[i]) != t[pidx].end()) pidx = t[pidx]
36 [s[i]];
37             else return 0;
38         }
39         return t[pidx].find('-') != t[pidx].end();
40     }
```

# 暨南大学本科实验报告专用纸(附页)

```
39     }  
40 };
```

## 5. 测试数据与运行结果

代码通过在线平台 `luogu.org` 正确性测试

测试点信息

源代码

测试点信息

Subtask #0

#2 AC 4ms/680.00KB	#3 AC 4ms/684.00KB	#4 AC 4ms/692.00KB	#5 AC 44ms/3.54MB	#6 AC 222ms/36.80MB	#7 AC 297ms/37.57MB	#8 AC 298ms/38.28MB
#9 AC 508ms/36.61MB	#10 AC 623ms/38.09MB	#11 AC 257ms/36.57MB				

Subtask #1

#1 AC 3ms/680.00KB
--------------------------



所属题目

P2580 于是他错误的点名开始了

评测状态

Accepted

评测分数

100

提交时间

2023-11-05 16:52:07