数据结构(上)

课程不允许录像，否则将追究法律责任，赔偿损失

九章算法强化班 第2章

扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: ninechapter
微博: http://www.weibo.com/ninechapter
知乎: http://zhuanlan.zhihu.com/jiuzhang
官网: http://www.jiuzhang.com

版权声明

九章的所有课程均受法律保护，不允许录像与传播录像
一经发现，将被追究法律责任和赔偿经济损失

Overview

数据结构：数据之间的关系，好的关系可以使得数据处理起来更加高效

1. Union Find 并查集

2. Trie 字典树

Union Find

并查集

一种用来解决集合查询合并的数据结构

支持O(1)find/O(1)union

并查集可以干什么？

1. 判断在不在同一个集合中。
   - find 操作

2. 关于集合合并
   - union 操作

并查集可以干什么？

1. 判断在不在同一个集合中。
   - find 操作

2. 关于集合合并
   - union 操作

并查集的操作

1. 查询 Find（递归？ 非递归？）

O(1) find

1. 合并 Union

O(1) union

两个操作时间复杂度证明:

https://en.wikipedia.org/wiki/Proof_of_O(log*n)_time_complexity_of_union%E2%80%93find

Log*n 的解释: https://en.wikipedia.org/wiki/Iterated_logarithm

查询 Find

- 模板代码

```
1 int find(int x) {
2     if (father[x] == x) {
3         return x;
4     }
5     return father[x] = find(father[x]);
6 }
```

```
1 int find(int x) {
2     if (father[x] == x) {
3         return x;
4     }
5     return find(father[x]);
6 }
```

比较有无路径压缩的区别

路径压缩：O(1) 找root的原因

合并 Union

- Key
  - 老大哥之间合并
  - 跟小弟没关系

```
1  public void union(int a, int b) {
2      int root_a = find(a);
3      int root_b = find(b);
4      if (root_a != root_b) {
5          father[root_a] = root_b;
6      }
7  }
```

并查集完整模板


```java
public class UnionFind{
    private int[] father = null;
    public int find(int x) {
        if (father[x] == x) {
            return x;
        }
        return find(father[x]);
    }

    public void union(int a, int b) {
        int root_a = find(a);
        int root_b = find(b);
        if (root_a != root_b)
            father[root_a] = root_b;
    }
}
```

Connecting Graph

http://www.lintcode.com/en/problem/connecting-graph/
http://www.jiuzhang.com/solutions/connecting-graph/

题意概括：n个节点的图，没有任何边存在。
有两个操作：
1.在a和b节点之间连上边
2.询问a和b节点是否连通

Connecting Graph II

http://www.lintcode.com/en/problem/connecting-graph-ii/
http://www.jiuzhang.com/solutions/connecting-graph-ii/

题意概括：n个节点的图，没有任何边存在。
有两个操作：
1.在a和b节点之间连上边
2.询问a所在的连通块的节点个数

Connecting Graph III

http://www.lintcode.com/en/problem/connecting-graph-iii/
http://www.jiuzhang.com/solutions/connecting-graph-iii/

题意概括：n个节点的图，没有任何边存在。
有两个操作：
1.在a和b节点之间连上边
2.询问当前图的连通块的个数

Connecting Graph问题的总结

- 并查集原生操作:
  - 查询两个元素是否在同一个集合内
  - 合并两个元素所在的集合

- 并查集的派生操作:
  - 查询某个元素所在集合的元素个数
  - 查询当前集合的个数

并查集实战例题

Google Interviewer: Number of Islands (九章算法班讲过)

www.lintcode.com/zh-cn/problem/number-of-islands

http://www.jiuzhang.com/solutions/number-of-islands/


Minecraft

Google Interviewer: Number of Islands II

http://www.lintcode.com/zh-cn/problem/number-of-islands-ii/

http://www.jiuzhang.com/solutions/number-of-islands-ii/

休息5分钟

课间思考：
给出几家合并公司的账号以及账号绑定的邮箱，
合并那些具有相同绑定邮箱的账号。

Example：
{ "A1": ["a1@gmail.com", "a2@gmail.com"],
"A2": ["b1@gmail.com", "a2@gmail.com"],
"A3": ["c1@gmail.com"],
"A4": ["c1@gmail.com", "d1@gmail.com"],
"A5": ["b1@gmail.com", "e1@gmail.com"]}
Output：["A1", "A2", "A5"], ["A3", "A4"]

Facebook Interviewer: Graph Valid Tree

http://www.lintcode.com/problem/graph-valid-tree
http://www.jiuzhang.com/solutions/graph-valid-tree/

Union Find O(n)

Surrounded Regions

http://www.lintcode.com/en/problem/surrounded-regions/

http://www.jiuzhang.com/solutions/surrounded-regions/

并查集考点

1、关于集合合并。

2、判断在不在同一个集合中。

解题你需要做的事：

把其他的操作，转化成这两件事情

Trie Tree

字典树

Trie树常考点

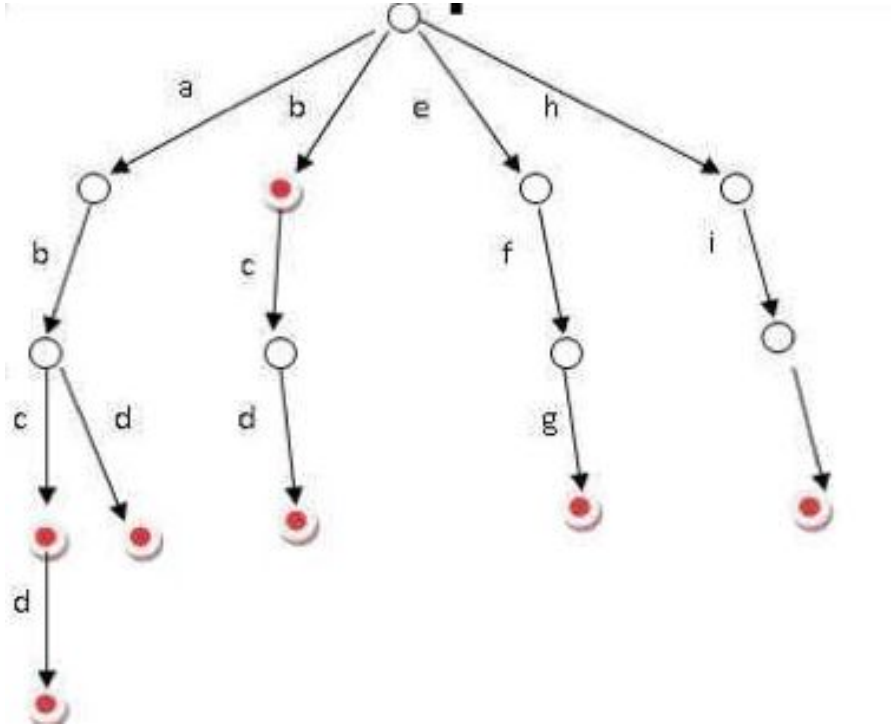
- Trie直接实现
- 利用Trie树前缀特性解题
- 矩阵类字符串一个一个字符深度遍历的问题

Snapchat Interviewer: Implement Trie

http://www.lintcode.com/en/problem/implement-trie/
http://www.jiuzhang.com/solutions/trie/

Implement Trie

- 假设有[b, abc, abd, bcd, abcd, efg, hii]这6个单词，查找abc 在不在字典里面



Hash vs Trie

时间复杂度Hash O(1) 是对于一个字符串

Trie树常考点

- Trie直接实现
- 利用Trie树前缀特性解题
- 矩阵类字符串一个一个字符深度遍历的问题

Snapchat Interviewer: Add and Search Word

http://www.lintcode.com/en/problem/add-and-search-word/
http://www.jiuzhang.com/solutions/add-and-search-word/

Trie树常考点

- Trie直接实现
- 利用Trie树前缀特性解题
- 矩阵类字符串一个一个字符深度遍历的问题

Trie树常考点

- Trie直接实现

- 利用Trie树前缀特性解题

- 矩阵类字符串一个一个字符深度遍历的问题(DFS+TRIE)
    - DFS 树 和 Trie树同时遍历

Microsoft Interviwer: Word Search II

http://www.lintcode.com/en/problem/word-search-ii/

http://www.jiuzhang.com/solutions/word-search-ii/

- Hash vs Trie做这道题目的区别
  - 把谁建成Trie树？

Word Search II

- Given a dictionary[aca, acc] and a matrix of upper alphabets,find all words in the dictionary that can be found in the matrix.
  - acaf
  - acad
  - acae

- 解题思路:
  - 把字典建成Trie树。
  - 用dfs的方法遍历矩阵，同时在Trie上搜索前缀是否存在。
  - 查询所有Trie里面有可能出现的字符。

Word Square

http://www.jiuzhang.com/solutions/word-squares/

对于数组["ball","area","lead","lady"]找到

可以组成的所有Word Square

[b a l l]

[a r e a]

[l e a d]

[l a d y]

Typeahead

搜索引擎

Trie考点

- 一个一个字母遍历
- 需要节约空间
- 查找前缀

Trie树常考点

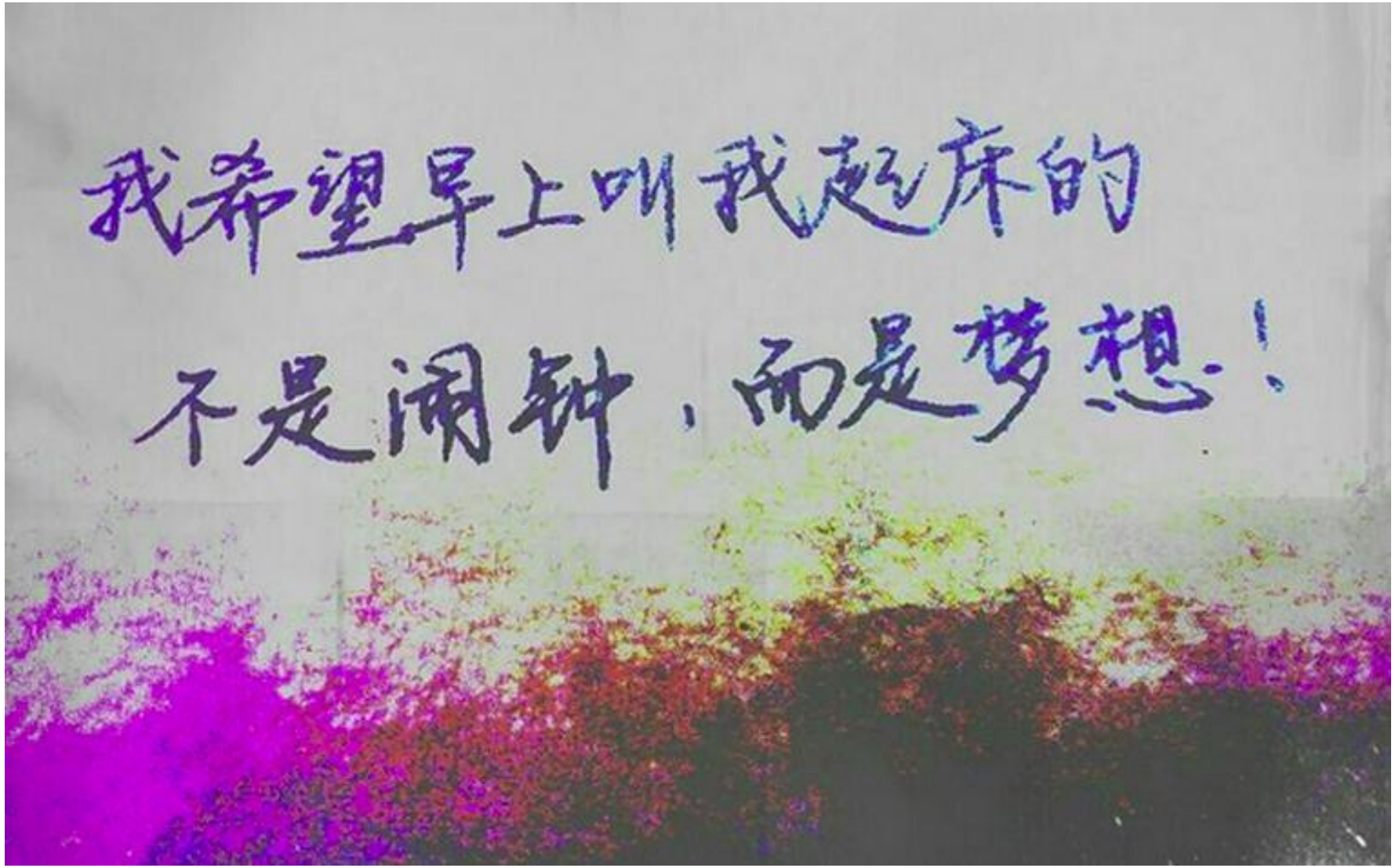
- Trie直接实现
- 利用Trie树前缀特性解题
- 矩阵类字符串一个一个字符深度遍历的问题
    - 充分活用前缀的性质

每日重点题

- Number of Islands II
  - 这道题充分体现了并查集的优势
- Implement Trie
  - 理解Trie的定义和实现
- Word Search II
  - Trie活用比较好的例子

Summary

- 数据结构的题目：
- Union Find: 集合合并，查找元素在集合里面。
- Trie: 一个字母一个字母查找，快速判断前缀。



Thank You