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

Overview

1. Union Find 并查集

2. Trie 字典树

3. Sweep Line 扫描线

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

并查集的操作

1. 查询 Find（递归？ 非递归？）

2. 合并 Union

查询 Find

- 模板代码

```
HashMap<Integer, Integer> father = new HashMap<Integer, Integer>();

int find(int x){
    int parent =  x;
    while(parent!=father.get(parent)) {
        parent = father.get(parent);
    }
    return parent;
}
```

合并 Union

- Key
  - 老大哥之间合并
  - 跟小弟没关系

```
HashMap<Integer, Integer> father = new HashMap<Integer, Integer>()

void union(int x, int y){
    int fa_x = find(x);
    int fa_y = find(y);
    if(fa_x != fa_y)
        father.put(fa_x, fa_y);
}
```

并查集完整模板


```
class UnionFind{
    UnionFind(){}
    HashMap<Integer, Integer> father = new HashMap<Integer, Integer>();

    int find(int x){
        int parent =  x;
        while(parent!=father.get(parent)) {
            parent = father.get(parent);
        }
        return parent;
    }

    void union(int x, int y){
        int fa_x = find(x);
        int fa_y = find(y);
        if(fa_x != fa_y)
            father.put(fa_x, fa_y);
    }
}
```

Find the Connected Component in the Undirected Graph

http://www.lintcode.com/en/problem/find-the-connected-component-in-the-undirected-graph/

http://www.jiuzhang.com/solutions/find-the-connected-component-in-the-undirected-graph/

连通块：无向图一个块中节点你找得到我，我也找得到你

有向图连通块

- 弱连通块
  - 有向图一个块中，你找得到我，我可以找不到你

- 强连通块
  - 有向图一个块中，你找得到我，我也找得到你

Find the Weak Connected Component in the Directed Graph

http://www.lintcode.com/en/problem/find-the-weak-connected-component-in-the-directed-graph/
http://www.jiuzhang.com/solutions/find-the-weak-connected-component-in-the-directed-graph/

路径压缩的查询 compressed_find

- 参考模板
- 平摊时间复杂度O(1)

```
int compressed_find(int x){
  int parent =  x;
  while(parent!=father.get(parent)) {
    parent = father.get(parent);
  }
  int temp = -1;
  int fa = x;
  while(fa!=father.get(fa)) {
    temp = father.get(fa);
    father.put(fa, parent) ;
    fa = temp;
  }
  return parent;

}
```

Google Interviewer: Number of Islands

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/

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。

Trie

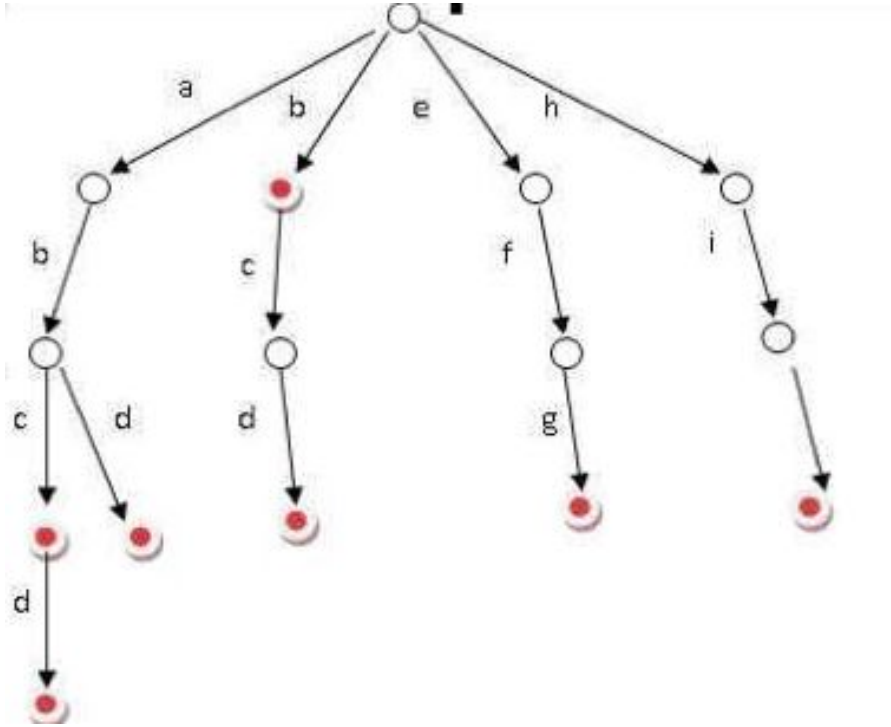
字典树

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

什么样的题目适合Trie？

- 一个一个字符串遍历

- 需要节约空间

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

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。

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/

Typeahead

搜索引擎

设计算法获得IP到城市的Map

http://www.jiuzhang.com/qa/262/

Trie考点

- 一个一个字符串遍历
- 需要节约空间
- 查找前缀

Sweep-Line

扫描线

Amazon Interviewer: Number of Airplane in the sky

http://www.lintcode.com/en/problem/number-of-airplanes-in-the-sky/

http://www.jiuzhang.com/solutions/number-of-airplanes-in-the-sky/

Number of Airplane in the sky

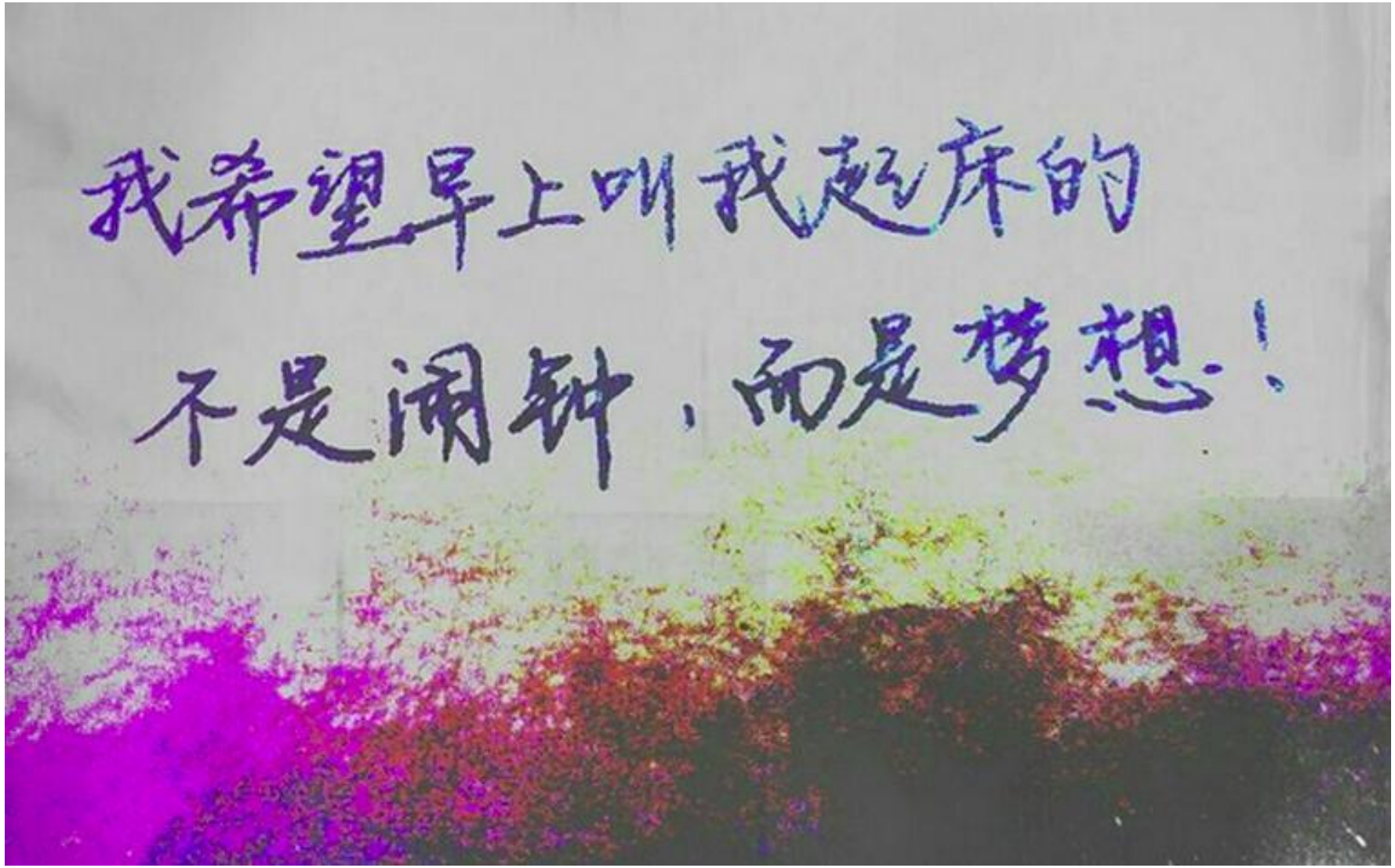
对于这道题有多少同学想过
按照“起点”或者“终点”对区间进行排序？

每日重点三题

- 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
- Number of Airplane in the sky
  - 扫描线入门题目

Summary

- 数据结构的题目：
- Union Find: 集合合并，查找元素在集合里面
- Trie: 快速找到一个元素，一个字母一个字母查找
- Sweep-line: 区间拆分



Thank You