

# 超现实树

清华大学 交叉信息研究院 李嘉图

清华大学 交叉信息研究院 杨天祺

# 题面简述

► .....

# 分数分布

- ▶  $= 100$ : 22人
- ▶  $\geq 40$ : 80人
- ▶  $\geq 20$ : 150人
- ▶  $< 12$ : 121人

自由吐槽

我咋觉得.....这题是 coRE-complete 的.jpg



可以大致理解成等价于  
停机问题的不可解问题

# 但没关系[呲牙]

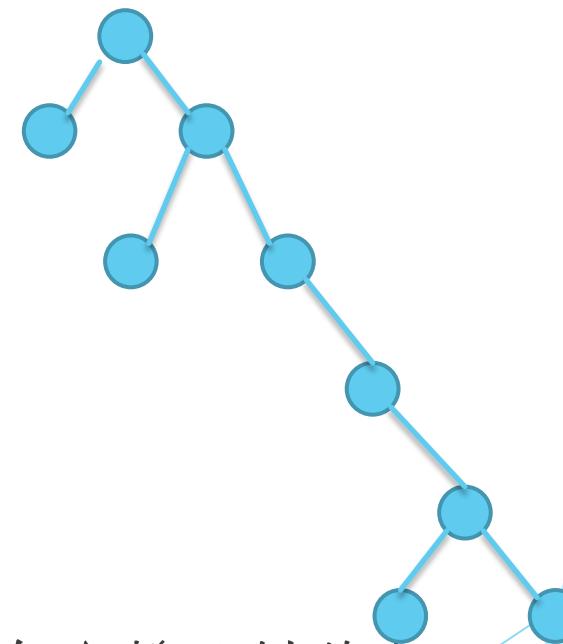
- ▶ 输出 Almost Complete 可以得到4分的好成绩！
- ▶ 输出样例可以得到12分的好成绩！

# 从零到一

- ▶ 假设  $H$  是所有给定的树集  $T$  的深度的最大值，那么枚举所有深度不超过  $H$  的树，判定它是否能够被生成
- ▶ 复杂度： $O\left(2^{2^H}\right)$
- ▶ 得分：20分

# 从一到二

- ▶ 仔细思考一下，发现我们只需要判断是不是所有深度恰好为  $H$ ，形如下面的树都能生成就行了，因为所有一般的树都能够扩展这种特殊的树的叶子节点生成
- ▶ 即由一条主链和一些可选的叶子节点构成
- ▶ 我们将其命名为树枝
  
- ▶ 这样就可以把复杂度优化到  $O(2^{2H})$  了
- ▶ 得分：32分
- ▶ 如果时间不够，性质二直接判一下是不是所有这样的树枝都有就行了，可以得到8分



## 性质三

- ▶ 由之前的性质（或者找规律），可以发现性质三的情况下，答案是YES当且仅当包含单个节点的树
- ▶ 得分：4分

# 我们得到一个算法了



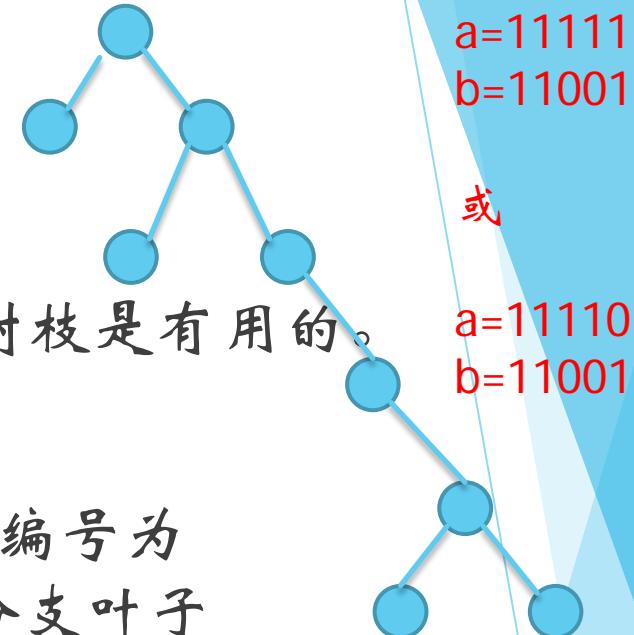
5分钟后.....

哦，我好像会线性的了



# 从二到无穷大

- ▶ 再思考一下发现，在给定的树集  $\mathcal{T}$  中，也只有树枝是有用的。
- ▶ 对于树枝，我们可以按照其**主链**和**分支叶子节点**编号为  $(a, b)$ ，其中  $a$  表示主链的方向， $b$  表示是否有分支叶子
- ▶ 可以将  $a$  和  $b$  分别建成两个 Trie，然后把每个树枝看成两个 Trie 上的一个点对
- ▶ 那么答案是 YES 当且仅当所有**叶子节点的点对**都能够被一个  $\mathcal{T}$  中某个树枝对应的点对**覆盖**



# 从二到无穷大

- ▶ 可以将  $a$  和  $b$  分别建成两个 Trie，然后把每个树枝看成两个 Trie 上的一个点对
- ▶ 那么答案是 YES 当且仅当所有叶子节点的点对都能够被一个  $T$  中某个树枝对应的点对覆盖
- ▶ 可以 DFS 一棵树，用线段树维护另一棵树的 DFS 序，维护每个点是否被覆盖了
- ▶ 复杂度： $O(n \log n)$
- ▶ 得分：100 分

# 但是我们要有梦想

- ▶ 事实上是可以做到线性的

编译选项

对于 C++ 语言	-lstd=c++11
注意事项	

# C++11? ? ?

- ▶ random      ✗
- ▶ regex          ✗
- ▶ unordered\_set    ✓

# 但是我们要有梦想

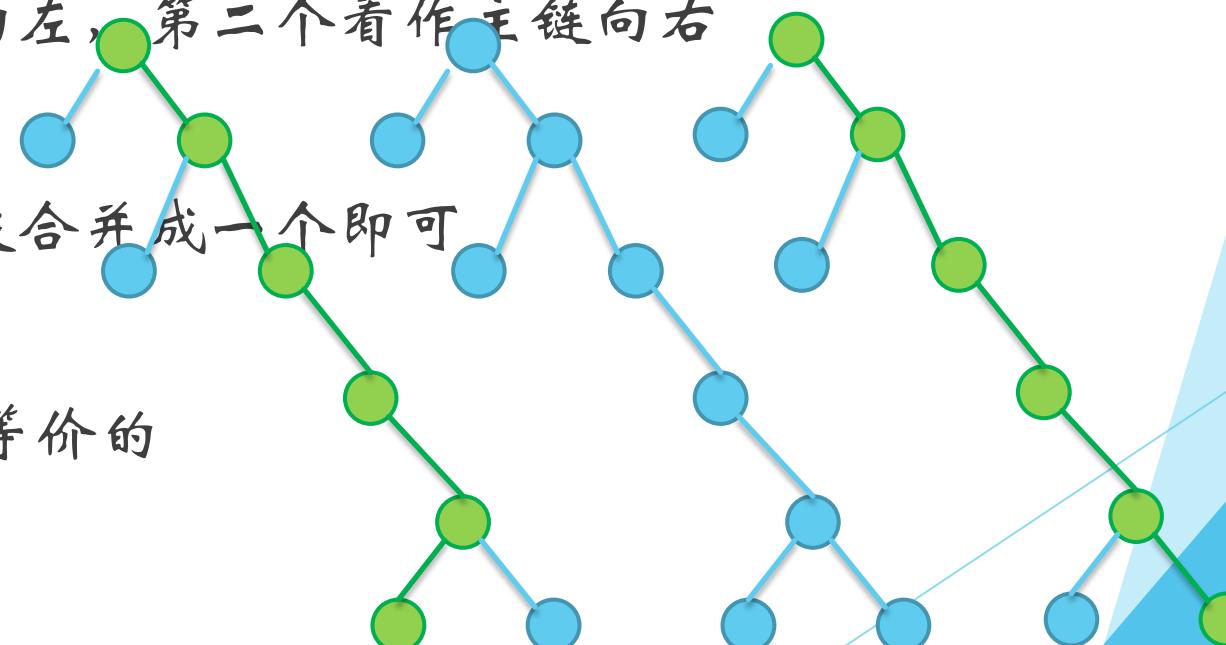
- ▶ 考虑一个合并点对的思路
- ▶ 假设  $(a.lc, b.lc)$ ,  $(a.lc, b.rc)$ ,  $(a.rc, b.lc)$ ,  $(a.rc, b.rc)$  四个点对都存在, 我们可以将其合并成一个点对  $(a, b)$ , 最终答案为 YES 当且仅当最终合并出  $(1, 1)$
- ▶ 按深度从大到小尝试合并, 用哈希表维护点对是否存在
- ▶ 复杂度:  $O(n)$
- ▶ 得分: 100 分

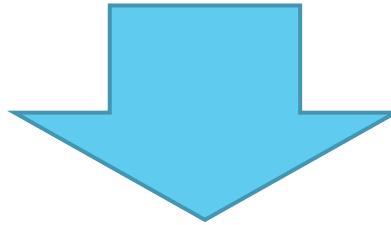
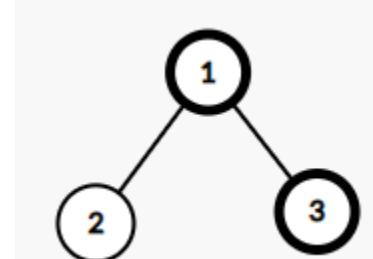
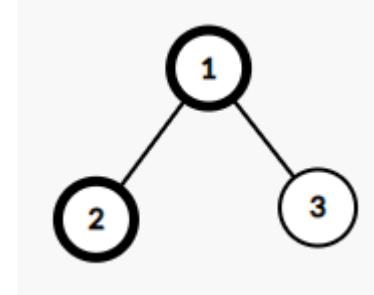
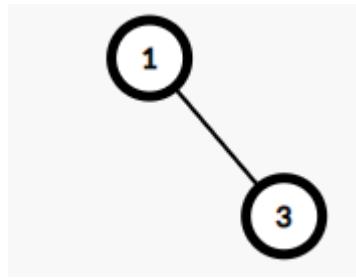
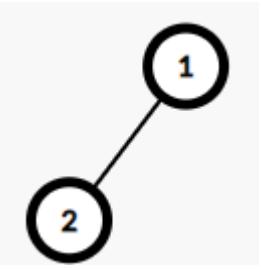
# 但其实并不需要哈希表.jpg

- ▶ 哈希表是一个w.h.p.的数据结构，在最坏情况可能会很差
- ▶ 其实用set就可以了。首先考虑为每一个深度分别维护set
- ▶ 注意到的关键性质是深度为  $h$  的树枝只有  $O(2^{2h})$  个，因此在这样一个set中查询的复杂度是  $O(h)$  的，但是这样一个树枝有  $\Omega(h)$  个节点，因此均摊一下复杂度就是线性的了
- ▶ 其实可以进一步证明总点数为  $n$  的树枝集合最多只有  $\frac{4n}{\log n} + \sqrt{n}$  个，证明可以自己尝试一下。

# 另一种视角

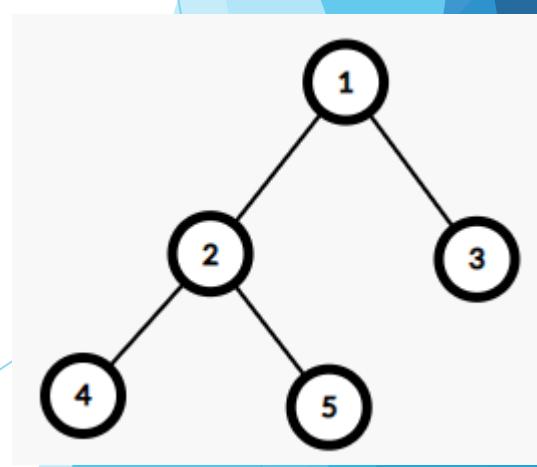
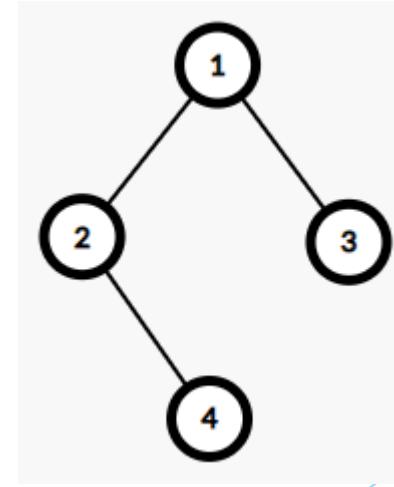
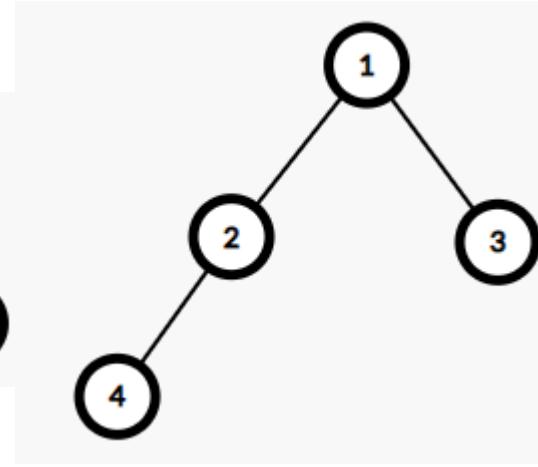
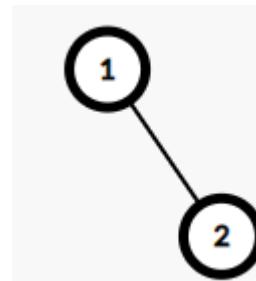
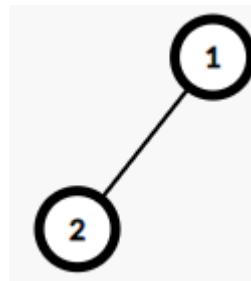
- ▶ 上述做法可以看成一个树合并的思想
- ▶ 我们把所有深度最深的中间节点有两个子节点的树枝拆成两个，第一个看作主链向左，第二个看作主链向右
- ▶ 每次将四种情况的树枝合并成一个即可
- ▶ 和之前 Trie 的思路是等价的

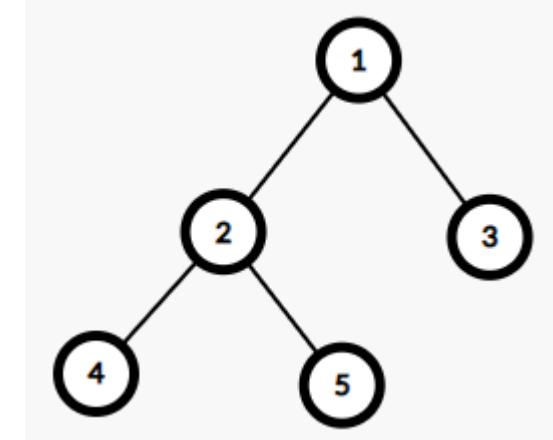
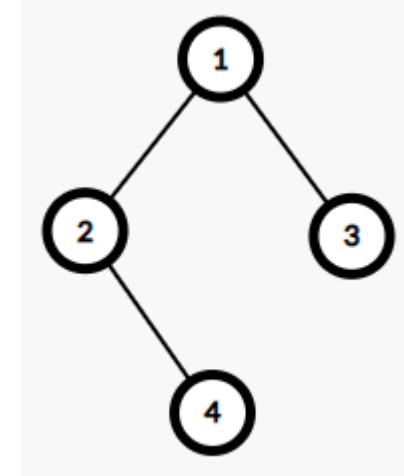
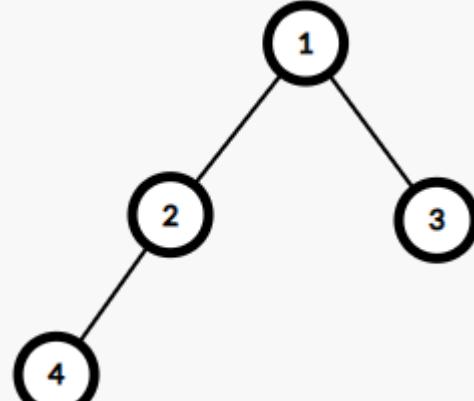
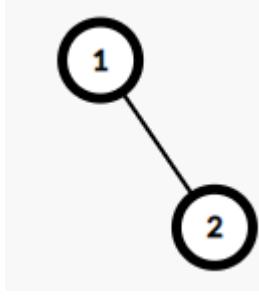
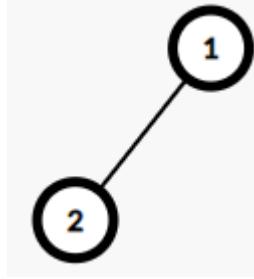




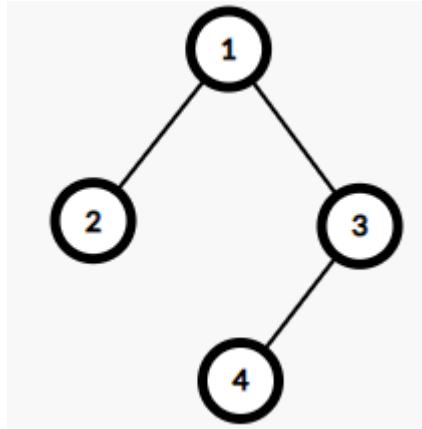
## 4缩1 or 3缩1

- ▶ 有不少选手想到了一个将三棵树缩成一个的做法，即不将深度最深的中间节点有两个子节点的树枝，然后把两种情况当成一种去做。
- ▶ 这样是错误的，反例：





不能替换成：

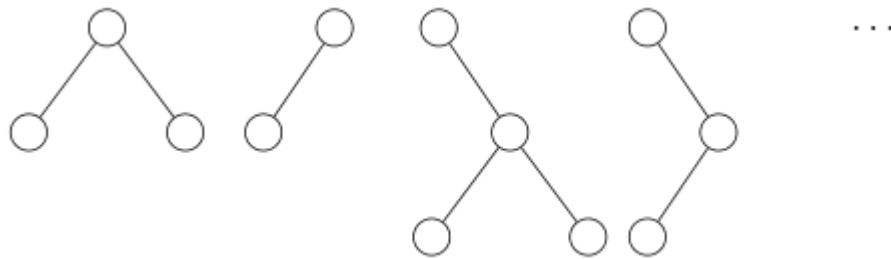


这个反例在大样例中有给出

这个错误的做法可以通过性质4的点

# 树的深度

- ▶ 答案为YES的总点数为  $n$  的集合中有用的树的最大深度是  $O(\sqrt{n})$  的



# To Conway

- ▶ 本题献给在新冠肺炎疫情中去世的数学家[约翰·康威 \(John H. Conway, 1937.12.26~2020.4.11\)](#)。他的研究兴趣涵盖组合游戏、群论等多个领域，在有限群分类、元胞自动机和组合游戏上做出了重要的贡献；他致力于数学科普，设计了曾风靡全球的“[康威的生命游戏](#)”（Conway's Game of Life）。
- ▶ 计算机科学大师高德纳（Donald E. Knuth）曾写过一篇对话体数学小说《[超现实数](#)》（Surreal Number）介绍康威为研究组合游戏而开发的一套纯数学理论。

Make Combinatorics Great Again!

欢迎提问