



春节十二响 (Spring)

ljt12138/清华大学计算机系 李嘉图

CommonAnts/清华大学计算机系 刘承奥

WerKeyTom_FTD/清华大学计算机系 王之栋



Staff



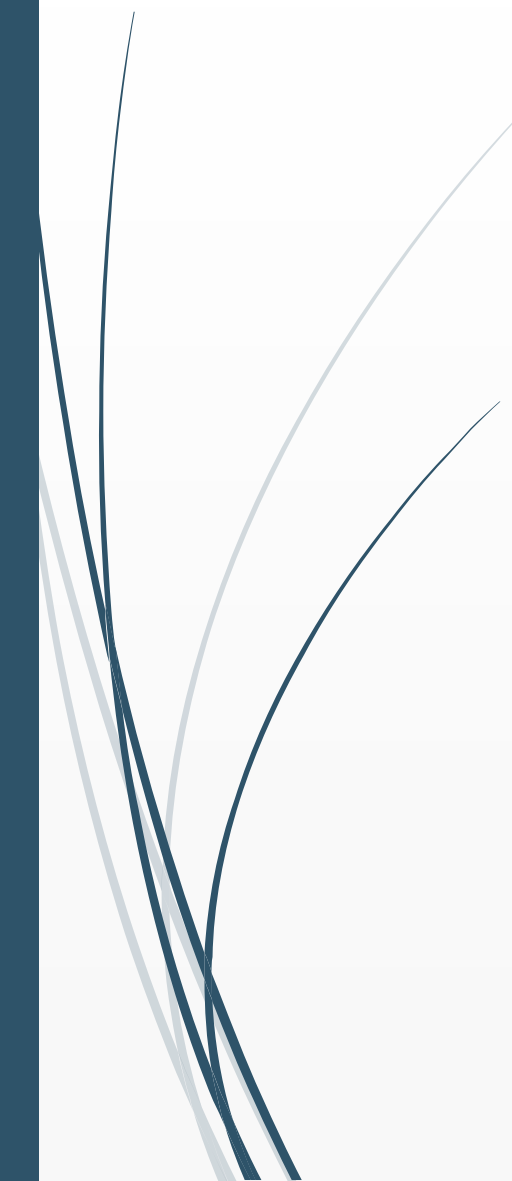
- ▶ Idea/Data/Statement: ljt12138
- ▶ Proof/Statement: CommonAnts
- ▶ Check: WerKeyTom_FTD

题意

- ▶ 给定一棵树，要求点集的一个划分，满足每一个划分块中都没有祖先-后代关系。
- ▶ 一个集合的权值是所有点点权的最大值，一个划分的权值是所有划分块的权值的和。
- ▶ 求最小权值划分方案。
- ▶ $n \leq 200000$ 。

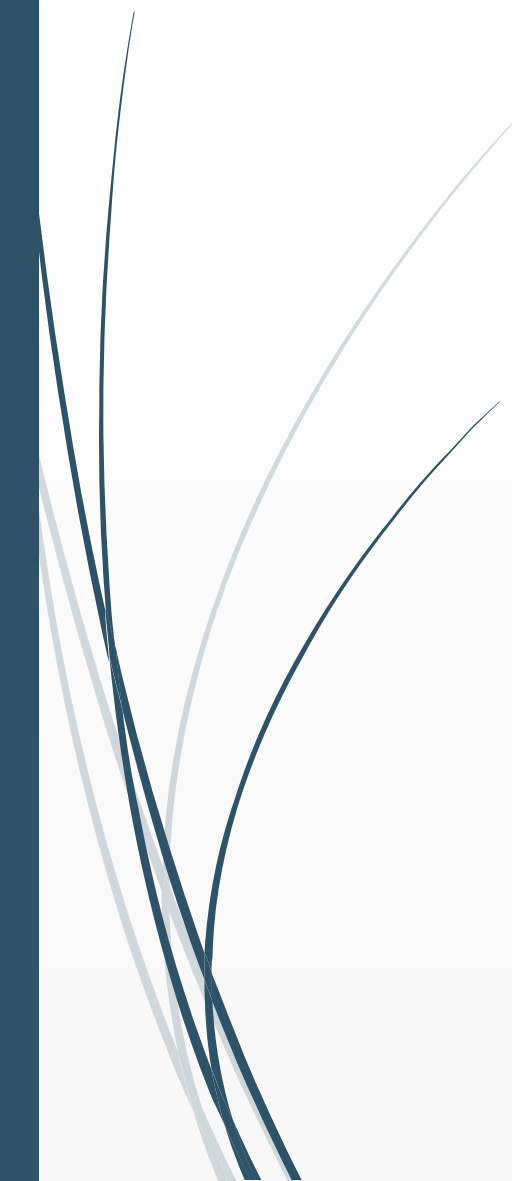


Subtask1: $n \leq 5$

- ▶ 手玩/爆搜
 - ▶ $O(?)$
- 



Subtask2: $n \leq 10$

- ▶ 不同的划分方案只有 $Bell(n)$, 可以暴力枚举每种划分方案。
 - ▶ 可以写一个漂亮的搜索来完成枚举。
 - ▶ $O(Bell(n))$
- 



Subtask3: $n \leq 16$

- ▶ 原问题可以用求一般图最大独立集的方法求。
- ▶ 状态压缩的动态规划，枚举子集转移。
- ▶ $O(3^n)$

Subtask4: 一条链

- 这个问题等价于：有两个独立运行的程序，程序A有 n 个模块，所需空间从大到小分别为 a_1, a_2, \dots, a_n ；程序B有 m 个模块，所需空间从大到小分别为 b_1, b_2, \dots, b_m 。同一个程序之间的两个模块彼此不能共享存储，不同程序的两个模块可以共享存储。
- 不失一般性，我们设 $n \geq m$ ，并令 $b_{m+1} = 0, b_{m+2} = 0, \dots, b_n = 0$ 以便进一步处理。那么很明显，对于程序A的每个模块，最多可以与程序B的一个模块共享一块储存空间，且一定会如此做。换言之，最优解是A中模块和B中模块的一个匹配。
- Claim: 最优方案一定是 a_i 和 b_i 匹配，即从大到小逐位匹配。
- Proof: 考虑 a_1 和 b_1 匹配的点，使用调整法将他们匹配，他们匹配的点匹配。



推广

- 为了将链的做法推广到树上，我们先证两个引理
- 

Lemma 1

- ▶ 设 M 是任意一个匹配方案, 其产生的 n 个模块序对所需的空间从大到小分别是 c_1, c_2, \dots, c_n 。
- ▶ 那么对于任意的 $1 \leq k \leq n$, 都有 $c_k \geq \max(a_k, b_k)$ 。

Proof

- ▶ 对序列长度施归纳法, 当 $n = 1$ 时是平凡的。
- ▶ 不失一般性假设 $a_1 \geq b_1$ 。
- ▶ 假设 a_1 与 b_{p_1} 匹配, 剩下的根据归纳假设有:

$$c_2 \geq \max(a_2, b_1) \geq \max(a_2, b_2)$$

$$c_3 \geq \max(a_3, b_2) \geq \max(a_3, b_3)$$

...

$$c_{p_1} \geq \max(a_{p_1}, b_{p_1-1}) \geq \max(a_{p_1}, b_{p_1})$$

$$c_{p_1+1} \geq \max(a_{p_1+1}, b_{p_1+1})$$

...

$$c_n \geq \max(a_n, b_n)$$

- ▶ 又有 $c_1 \geq a_1 = \max(a_1, b_1)$ 。因此有 $c_i \geq \max(a_i, b_i)$ 。

Lemma2

- ▶ 对于多条链的情景，结论也正确，即：

$$c_i \geq \max(a_{1,i}, a_{2,i}, \dots, a_{k,i})$$

- ▶ 其中 a_1, a_2, \dots, a_k 为若干条链。 $a_{i,*}$ 按照从大到小排序。

Subtask5: $n \leq 2000$

- ▶ 前面的引理说明，当我们按照从大到小逐位对齐，不仅是最优的，而且逐位优于任何一个方案。
- ▶ 那么我们考虑在树上合并，合并两个子树的答案，只需将所有划分按权值从大到小排序，然后逐位对应即可。
- ▶ Proof? 方案逐位优于任何一个方案，合并之后也有这个性质。

Theorem

- 树上合并算法是正确的。
- 考虑证明加强结论，说明树上合并算法得到的方案逐位优于任何方案。
- 对树的高度做归纳，当树高为1时是平凡的。
- 当树高 > 1 时，根据前面的引理和归纳假设，我们将子树两两合并起来的结果仍然是逐位优于任何方案的。

- Proved by CommonAnts.
- 这样只要暴力合并，复杂度就是 $O(n^2)$ 的。

Final: $n \leq 200000$

- ▶ 上面的合并算法明显可以优化。
- ▶ 考虑一个子树划分出的集合个数一定是子树深度，而合并的复杂度是合并双方长度的最小值。
- ▶ 对于一个节点 nd ，找到其最深的孩子 son ，将其他孩子合并到 son 上，最后把根节点加入进去。
- ▶ 总的合并次数是 $O(n)$ 的，使用一个 set 或 $priority_queue$ 维护均可。
- ▶ 复杂度 $O(n \log n)$ 。

Story

- 题目是操作系统课上想出来的。
- 内存覆盖曾经是真实存在的技术，1970s的IBM的机器上提供了一种特殊的覆盖指令，使得程序可以做类似这样的操作。但细节有所不同。
- 当时用于实用的一般图自动化内存覆盖算法是：
 - 缩点
 - 将入度超过2的节点和支配点合并（不能覆盖）
 - 树上覆盖
- 但是由于页式内存管理技术的发展，内存覆盖的需求几乎消失了.....

Story

- ▶ 本来题目是想叫做清明十二响的，后来想想有点不对。
- ▶ 本来题目是和IBM5100相关的，后来为了和Hope统一就改成了现在的样子。
- ▶ 题面是由 Σ^* Studio提供的
 - ▶ 同期作品还有快咕咕子的AnotherBetaCat同学的OI Diary系列新作
 - ▶ 往期作品包括WC 2019相声《选班长》(OI Diary Monitor)
 - ▶ 未来作品可能还有未经过平衡性测试的桌游 OI Diary Fantasy
 - ▶ 文体两开花，请大家多多关注
- ▶ 希望十二省联考的各位没有十二响。



THX

➡ 欢迎提问

