

Trie树习题精讲

——POJ3764



主讲人：邓哲也



POJ3764

- 题目描述
 - 给出 n 个节点的树，每条边有边权 w 。
 - 请你找一条路径，使得这条路径上所有边权的异或和最大。
- 数据范围
 - $n \leq 100,000$, $0 < w < 2^{31}$
- 样例
 - 输入
 - 4
 - 0 1 3
 - 1 2 4
 - 1 3 6
 - 输出
 - 7

POJ3764

- 暴力的做法：
 - 枚举每个点当根
 - dfs整颗树，获得根到每个节点的路径异或和s[]
 - 时间复杂度 $O(n^2)$

POJ3764

- $x \text{ xor } x = 0$
- 一个数异或两次会抵消掉。
- x 到 y 的路径异或和就等于 $s[x] \text{ xor } s[y]$!
- 为什么?

POJ3764

- 这样只要任取一个点为根，dfs得到s[]数组后，枚举x，y即可
- 还是 $O(n^2)$

POJ3764

- 现在的问题是，给定一个 $s[]$ 数组，找两个元素，使得他们的异或和最大。
- 给一个数 x ，在数组 s 中找一个数 y 使其和 x 的异或和最大。
- 涉及二进制的题，我们可以选择按位来考虑。

- 假设x的二进制有31位，我们把它二进制写出来，最高位称为第30位，最低位称为第0位。
- 因为要让异或值最大，所以我们从高位开始考虑。
- 如果x的第30位是1 —— `if (x >> 30 & 1)`
- 那我们一定希望y的第30位是0，这样异或之后的第30位是1。
- 否则就算第29~0位异或后都是1，也比 2^{30} 小。
- 同理，如果x的第30位是0，我们希望y的第30位是1。

POJ3764

- 运用贪心的思想，假设现在枚举到 x 的第 i 位
- if x 第 i 位是1 我们要尽可能找第 i 位为0的 y
- if x 第 i 位是0 我们要尽可能找第 i 位为1的 y

POJ3764

- 先把所有的数的二进制当作字符串，存进trie中。
- 从高往低枚举二进制位时，设置一个指针p，一开始指向trie的根节点。
- if x第i位是1 我们要尽可能找第i位为0的y
 - 如果p->go[0] != NULL 说明存在第i位为0的数，我们就让p往p->go[0]节点走，答案中的第i位为1。
 - 如果p->go[0] == NULL 说明不存在第i位为0的数，我们只能退而求其次，让p往p->go[1]节点走，答案中的第i位为0。
- if x第i位是0 我们要尽可能找第i位为1的y
 - 同理。

代码实现

- `int findans(int val) {`
- `int x = root, ans = 0;`
- `for (int i = 30; i >= 0; i --) {`
- `int k = (val >> i & 1) ^ 1;`
- `if (t[x].go[k]) {`
- `x = t[x].go[k];`
- `ans += 1 << i;`
- `} else {`
- `x = t[x].go[!k];`
- `}`
- `}`
- `return ans;`
- `}`

下节课再见