

新年的繁荣 解题报告

安徽师范大学附属中学 吴作凡

1 试题来源

原创。UOJ Goodbye Yiwei的C题，链接：<http://uoj.ac/problem/176>

2 试题大意

一个完全图有 n 个点，第 i 个点的点权是 a_i ，满足 $0 \leq a_i < 2^m$ 。第 i 个点和第 j 个点之间边的边权是 $a_i \text{ and } a_j$ ，求最大生成树大小。

共分为6个子任务，具体范围如下：

子任务编号	分值	n	m
1	15	$n \leq 1000$	$m \leq 18$
2	15	$n \leq 5000$	$m \leq 18$
3	10	$n \leq 10^5$	$m \leq 1$
4	15	$n \leq 10^5$	$m \leq 12$
5	15	$n \leq 10^5$	$m \leq 15$
6	30	$n \leq 10^5$	$m \leq 18$

3 算法介绍

3.1 算法一

对于子任务1，直接把完全图建出来跑kruskal算法就好啦。

时间复杂度： $O(n^2 \log n)$ ，期望得分15分。

3.2 算法二

对于子任务2，依然可以把完全图建出来，但是使用prim算法可以降低复杂度。

如果继续使用kruskal算法，可以发现瓶颈在于排序，注意到边长不会超过 2^m ，于是可以用基数排序来降低复杂度。

时间复杂度： $O(n^2)$ 或 $O(n^2\alpha(n))$ ，期望得分30分。

3.3 算法三

对于子任务3，点权为0或1，只有两个1相连的时候才会有1的贡献，那么最优方案就是把所有的1全部连起来，答案就是1的个数-1，注意特判没有1的情况。

时间复杂度： $O(n)$ ，结合算法二期望得分40分。

3.4 算法四

对于子任务4，考虑kruskal算法，我们可以发现肯定存在一种方法，使得每个点连出去的某条最大边都被选择，由于 $a \text{ and } b \leq a$ ，所以对于相同权值的点我们可以先连好并缩点，这样最多只剩下 2^m 个点了，再使用算法二就可以通过这些部分分。

时间复杂度： $O(4^m)$ ，结合算法二期望得分55分。

3.5 算法五

对于子任务5，顺着刚才的思路，如果 $a \text{ and } b = a$ ，那么肯定可以先将 a 和 b 用一条边连起来，因为 a 的二进制的1被 b 完全包含，那么剩下的点连到 a 上肯定都不会比连到 b 上去优，就可以把 a 删去了。最后留下的所有的点都互不包含，这样的点数最多是 $\binom{m}{m/2}$ 的（也就是所有二进制含有 $m/2$ 个1的数），那么在这些点上跑算法二就可以了。

时间复杂度： $O(\binom{m}{m/2}^2)$ ，结合算法二期望得分70分。

3.6 算法六

来考虑一下普通的最大生成树除了kruskal和prim还有什么算法。既然可以选择最大边，那么我们可以对于每个点都选择最大边连边，缩完联通块，再对于每个联通块选择最大边继续缩，缩了 i 次以后每个联通块大小都至少是 2^i ，所以只要缩 $\log n$ 次就行了（这个算法叫作Boruvka算法）。

那么现在的问题就是这样，有 n 个数，每个数有一个颜色，对于每个数 a 求出一个异色的数 b ，使得 a and b 最大。这里我们可以使用trie或者子集和变换。

如果把and改成xor我想大家都会用trie来做，先把每个数插入trie中，维护每个点的子树中颜色的最大值和最小值，这样就能知道是否有除了当前查询的颜色以外的其他颜色了，询问就直接跑一跑就好了。

这里是and，如果当前询问的数的这一位是0，那么两个子树都需要跑，时间复杂度就不对了。但我们发现永远不会只需要跑0子树，而不跑1子树，那么我们就可以先建出trie，然后从下到上依次将1子树合并到0子树上去，这样每个点的左子树就变成了0+1，右子树还是1，询问到0就到左子树，到1就跑右子树，于是询问就变成 $O(m)$ 了。建树的时候可以发现每个点被它的每个祖先都要访问一次，所以复杂度是 $O(m2^m)$ 的。

可以发现之前的合并以后每个点代表着它的子集，于是可以把trie改成子集和变换，关于子集和变换可以看吕凯风学长的2015年国家集训队论文《集合幂级数的性质与应用及其快速算法》。复杂度和trie相同。

时间复杂度： $O((n + 2^m)m \log n)$ ，期望得分100分。

3.7 算法七

我们继续考虑使用kruskal算法。首先我们使用算法四的思想去重，接着我们可以从大往小枚举边权 p ，存在and超过 p 的点都已经在同一个连通块了，那么最多有 m 个连通块需要连 p 的边，用并查集维护一下。

实现的时候可以用一个数组 a_p 表示缩完边权为 p 的点以后形成的连通块，连边的时候就把 a_p or 2^i ($0 \leq i < m$)拿出来就好了。

时间复杂度： $O(m2^m \alpha(n))$ ，期望得分100分。

3.8 算法八

我们扔掉普通的最大生成树算法，也可以利用分治来解决这个问题。

依然是不断地合并连通块，不过现在是按照二进制位从高到低合并，对于每一个连通块维护一个trie。考虑第 k 位，将所有具有1子树的连通块的1子树都取出来分治到 $k-1$ 层，然后将这些块合并，和剩下的块一起分治到 $k-1$ 层。

考虑时间复杂度，共有 m 层，但是一个点可能在同层出现多次，那么复杂度上界应该相当大，但是实际运行起来相当快，甚至可以通过 $m=30$ 的数据，期望得分100分。

4 参考程序

- andmax1.cpp是算法六。
- andmax2.cpp是算法七。
- andmax3.cpp是算法八。