位运算最小生成树

目录

Ⅰ. 问题定义

有 n 个点，第 i 个点有点权 a_i，$0 \leqslant a_i < 2^m$。边 E(i,j)[1]的边权为 $f(a_i,a_j)$。f(a,b)表示一个位运算，一般为 AND 或者 XOR。求这张图的最大/最小生成树。[2]

Ⅱ. 基本解法

除边权跟位运算有关，这类题目说到底还是 MST 问题，座椅可以暴力建出这些边，然后跑 Prim 或者 Kruskal。但因为边数可

[1] 任意两点之间都存在边，所以这是一张完全图
[2] 注意！边权与边权之间是相加，不再是位运算

能会达到 $O(n^2)$的级别，所以该种解法局限性非常大

Ⅲ. 通解通法

一. 基本思路

因为是 MST 问题，所以可以考虑 Borůvka 算法[3]。所以现在的问题转换为：给每一个值分配一个颜色，给定一个值 a_x，如何在$\{a_i\}$中找到一个异色的 a_y使 $f(a_x,a_y)$最大。

二. 对于 XOR

因为对于位运算，二进制下每一位相对独立，所以考虑 Trie。

对于每一轮连边，我们新建一棵 Trie，将每一个数转化成二进制，然后插入到 Trie 中。同时在 Trie 中，记录每一个点的子树中颜色的最大值和最小值。

然后对于每一个点，直接在 Trie 中查询即可

三. 对于 AND

基本思路与 XOR 相同，但是如果我们正在查询 ai，其第 j 位为 0，这样的话，0/1 这两棵子树我们都需要进入。这样的话，时间复杂度就没有保障了。

所以我们在建好 Trie 树后，还需要把所有 1 的子树给合并

[3] 不了解 Borůvka 的话，可以去看看 wikipedia 或者在我的 blog 上搜一搜

到 0 的子树中去。

这样的话，无论情况如何，我们都只用进入一棵子树了

四. 时间复杂度分析

运行 Borůvka 算法时需要 log（n）波连边

每一次连边需要建一棵 Trie 树，每一次建树复杂度为 O(n*m)

之后需要使用并查集来连每一条边，复杂度为 O(α(n)*n)

所以总时间复杂度为 O（n*m*log(n)）[4]

五. 代码参考

①. XOR 版本：http://paste.ubuntu.com/23080635/[5]

②. AND 版本：http://paste.ubuntu.com/23080636/[6]

Ⅳ. XOR 的特殊解法

因为 XOR 版本的特殊解法[7]在时间复杂度与编程复杂度上都劣于通解通法，所以不提供相关资料。

---

[4] AND 版本为 O((n*m+2^m)*log(n))，因为需要 Merge()

[5] 配套例题：http://www.lydsy.com/JudgeOnline/problem.php?id=3943

[6] 配套例题：http://uoj.ac/problem/176

[7] 即分治算法

Ⅴ. AND 的特殊解法

一. 基本思路

从大到小枚枚举边的权值，从贪心的角度来讲，如果可以连接该边，那么直接连接就可以了。所以我们得出了一种 $O(n*2^m*\alpha(n))$的算法

二. 优化思路

首先，权值相同的点是等价的，所以可以缩起来

所以我们可以权值而不考虑点了

再配合一点代码的 Trike[8]，我们可以把时间复杂度优化到 $O(m*2^m*\alpha(n))$

三. 代码参考

AND 最大生成树版本：http://paste.ubuntu.com/23080662/[9]

Ⅵ. 参考文献

[1]Wikipedia : https://en.wikipedia.org/wiki/Bor%C5%AFvka%27s

[2]UOJ 题解： http://vfleaking.blog.uoj.ac/blog/1244

[8] Trike 这种东西，语言不好描述，推荐看看代码

[9] 因为没有找到 AND 最小生成树的题目，所以不提供参考代码

[3]《一些杂七杂八的东西》 吴作凡著[10]

[10] 可以在这里下载：http://pan.baidu.com/s/1c2keuZ2