

## 4 QOJ1844 Cactus

### 4.1 题目大意

给定一棵  $n$  个点,  $m$  条边的仙人掌, 其中仙人掌表示每条边至多属于一个简单环的无向连通图。可以对它进行若干次操作:

1. 选择一个度数为奇数的点, 将所有与它相邻的边删去。
2. 建立一张新的图, 有  $2n$  个点和  $2m + n$  条边。其中对于原图的每条边  $(u, v)$ , 在新图上有边  $(u, v)$  和  $(u + n, v + n)$ 。且对于所有  $1 \leq i \leq n$ , 在新图上有边  $(i, i + n)$ 。

第二种操作只可以使用至多 1 次, 第一种操作使用次数不限。要求构造一个操作序列, 使得最终剩余的边数尽可能少。

### 4.2 数据范围

$n \leq 3 \times 10^5, n - 1 \leq m \leq \frac{n(n-1)}{2}$ 。保证给定的图无自环、重边, 且是一个仙人掌。

### 4.3 解题过程

从简单情形入手考虑, 先假设给定的仙人掌是一个环, 考虑最少最终会剩余多少条边。此时所有点的度数都是 2, 即必须先进行一次 2 操作。如果  $2 \mid n$ , 则在这张新图上, 我们依次操作点  $2, 4, 6, \dots, n$ , 以及  $3 + n, 5 + n, 7 + n, \dots, 2n - 1$ , 最后操作  $1 + n$ 。如果  $2 \nmid n$ , 我们依次操作  $2, 4, 6, \dots, n - 1$ , 以及  $3 + n, 5 + n, 7 + n, \dots, 2n$ , 最后依次操作  $n, 2 + n, 1 + n$ 。

容易看出, 对于任意大小的  $n$ , 上述操作流程每次操作的点度数一定为 3 或 1, 即操作一定合法, 并且最终会把所有边都删空。进一步地, 最后一步中一定删除了恰好一条边  $(1, n + 1)$ 。

回到原题, 我们每次找出一个环, 使得这个环仅有一个点同时位于其它环中。记这个点为  $x$ , 那么可以将它看作上述流程中的 1 这个点。则进行上述流程直到最后一步, 此时还剩下恰好一条边  $(1, 1 + n)$ , 即原图中的  $(x, x + n)$  这条边。那么对于除了这个环以外的其它环, 其形态被完整地保留, 所以可以看作将这个环删去了, 并递归求解剩余的子问题。

然而这个做法有一个问题, 即在解决单个环的问题时, 1 的度数是偶数。但是现在可能会出现 1 的度数是奇数的情形, 此时直接使用上述方法是不可行的。

对于这个问题, 我们先在最开始的图上进行一些操作, 每次任意选择一个度数为奇数的点对它进行操作。这样一直操作到不能操作为止, 此时所有点的度数都是偶数。下面我们还需要证明在这样的一张图上任意进行删环的操作, 所得到的图仍然满足所有点的度数都是偶数。

先证明一个引理:

**引理 4.1.** 一个仙人掌上所有点的度数均为偶数当且仅当这张图上不存在割边。

证明. 如果一个仙人掌上没有割边, 说明每条边属于恰好一个简单环中。则对于每个点, 其所有邻边都属于恰好一个简单环。由于一个简单环中有恰好两条边与每个点相邻, 所以若一个点属于  $k$  个简单环, 其一定有恰好  $2k$  条邻边, 即度数为偶数。

下面证明, 若一张图上存在割边, 则至少有一个点度数为奇数。

对点数归纳,  $n \leq 3$  的情况显然。下面假定对所有  $n' < n$ , 都已经证明上述结论, 尝试推出  $n$  时也成立。

取出所有割边, 剩余的图被分成若干连通块。将一个连通块看作一个点, 根据割边的定义, 它们由割边连接形成一棵树。则任取一个叶子所代表的连通块, 其中的点数  $n' < n$ 。根据归纳假设, 由于其中不存在割边, 所以每个点的度数均为偶数。

另一方面, 对于将这个叶子与其它连通块连起来的唯一割边  $(u, v)$  (其中  $u$  在连通块内),  $u$  连向连通块外有且仅有这一条边。因此  $u$  在原图中的度数为奇数, 即存在一个度数为奇数的点。

□

回到原题, 根据引理, 现在这张图上不存在割边。而我们每次删除一个仅有一个点与其它环相交的环, 不会产生任何割边。即每次操作之后得到的图均没有割边, 故所有点的度数均为偶数。

最后只需要能每次找到一个仅有一个点位于其它环中的简单环即可。可以建出原图的圆方树 [BCC], 则每次选择深度最深的方点, 其所代表的环就满足上述要求。删去这个环后, 不会改变圆方树的形态, 所以直接按照深度从大到小依次选择所有环即可。

这样做只需要在一开始执行一次 2 操作, 且每个点只会被 1 操作操作至多 1 次, 即总操作次数  $O(n)$ , 时间复杂度也为  $O(n)$ 。