

Railway 解题报告

长沙市一中 张天扬

1 试题来源

POI2010

<http://www.lydsy.com/JudgeOnline/problem.php?id=2080>

2 试题大意

给定 n 以及一个 $1 - n$ 的排列。求它能否使用两个栈完成排序。

$n \leq 10^5$

3 算法介绍

3.1 暴力

双栈排序是一个经典的问题。

我们考虑两个元素 $a_i, a_j (i < j)$ 在什么情况下不能放入同一个栈中。

结论是：如果存在一个数 k ，满足 $i < j < k$ 且 $a_k < a_i < a_j$ ，那么 a_i 和 a_j 不能放入同一个栈中。

显然，这是因为：如果把他们放入同一个栈中，那么在放入 a_k 前， a_i 和 a_j 都必须入栈且还没有弹出。而 $i < j$ ，因此 a_i 会在 a_j 前入栈。那么 a_i 不可能在 a_j 之前弹出。这与最后的序列要有序矛盾。因此 a_i 和 a_j 不能放入同一个栈中。

而如果我们得到了所有的形如“ a_i 和 a_j 不能放入同一个栈中”的信息，那么我们只需要判断得到的图是不是一个二分图即可。如果是二分图，把二分图的两部分别放入两个栈中就可以了。

这样我们就得到了一个 $O(n^3)$ 的暴力算法。这个算法可以通过记录后缀最小值来优化到 $O(n^2)$ 。

3.2 标准做法

注意到：如果我们像上面一个做法一样把整个图求出来的话，图中的边数可能达到 $O(n^2)$ 级别。那么这显然是不行的。

我们不妨从另一个方向来考虑：我们得到二分图的目的，是求出图的一个2-染色。那么我们不妨假设原图是一个二分图，求出它的一个2-染色，之后再按题意模拟一遍，看能不能完成排序就可以了。

那么我们维护一个数据结构，保存当前所有还没有被染色过的点。我们考虑一个当前已经被染色过的点，我们要在数据结构中找到一个点与当前点间有边。重复这个操作直到没有点与当前点有边为止，然后再考虑另一个已经被染色的点。如果所有被染色的点都考虑了一遍，就从没有染色过的点里随便拿出一个。

那么我们需要的数据结构是：能找到一个与给定点间有边的点并把它删掉。分情况讨论之后容易知道，我们可以用两棵线段树来维护。

复杂度是 $O(n \log n)$ 。可以通过本题。

3.3 复杂度更加优秀的算法

考虑上面的做法为什么是对的。最重要的一点是：对于一个二分图，如果我们删掉其中的一些边，只要图的连通性没有改变，那么图的2-染色也不会改变。于是我们的思想是：连尽量少的边，使得图的连通性与把边全部连完的连通性一样。上面的做法只连了 $O(n)$ 的边，已经足够优秀，但是每连一条边需要 $O(\log n)$ 的时间。下面介绍一种 $O(n)$ 的算法。

我们按照数值从大到小依次考虑每一个数，维护一个指针 R 表示当前还没有考虑的数中位置最靠后的一个。维护一个单调栈，栈中每个元素是当前它所在的连通块中值最小的点，栈中元素的位置依次递增。

考虑当前数 x 。有两种情况：

1.如果 x 位于指针 R 的位置上，那么我们就不连任何边，只把 R 左移直到第一个还没有考虑的数。

2.否则，我们从栈的顶端元素开始依次考虑：如果它的位置大于 x 的位置，那么我们就把它弹出来。如果弹出来了，且它的位置小于 R ，那么说明它和 x 间是有边的，把它们间连一条边即可。最后，把 x 放入单调栈中。

只连这些边是不够的，因为我们实际上是用权值最小的元素来“代表”一

个连通块，而这有可能会产生遗漏，造成连通性错误。解决的方法也很简单，对每一个元素，如果它后面第一个大于它的元素与它之间应该有边，就连一条边。这样就能够保证连通性正确了。

复杂度是 $O(n)$ 的，达到了理论下界。