

世界线 解题报告

宁波市镇海中学 邹逍遥

1 试题来源

UOJ Round #10

<http://uoj.ac/problem/153>

2 试题大意

有一个长度为 n 的全排列 a ，你需要求出它的值。

你可以进行两轮操作，每轮操作可以建立一张图，然后进行 k 轮询问，每次询问给出 x, y ，交互库会回答你 a_x 和 a_y 的连通性。

在正确求出 a 的的基础上最小化两轮操作 k 的和。如果怎么样都无法确定返回-1。

3 数据范围

$n \leq 10000$

两轮操作中 k 的和不超过 2×10^6 视为正确。

4 算法介绍

4.1 算法一

先尝试一下小数据。 $n = 1$ 不需要询问， $n = 2$ 无解。

$n = 3$ 和 $n = 4$ 时只需要第一次 $1 - 2$ 第二次 $2 - 3$ ，就能区分出所有点。

观察一下 $n = 3$ 和 $n = 4$ 的解，发现这种解可以拓展到 n 更大的情况。即连成一条链，先判断出最边上的点，然后顺着往里推推出剩下的点。

由于一个点只会和另外的一个点连边，所以每个点期望要询问 $O(n)$ 次才能鉴定出来，那么询问次数为 $O(n^2)$ 。只能做到 n 为1000左右。

4.2 算法二

如果想减少询问次数，其实就是减小块数。因为询问的复杂度是块数 $\times n$ 的（只需要对于每个块扫一遍所有的点即可）。

由于两次块数之积不能小于 n （否则两个点两次都在同一块就无法区分了），所以块大小之和最小只能到根号范围。

那么重点就在于怎么区分同一个分组中的块。使用块大小来判断是一个不错的办法。

假如 n 满足 $n = \frac{k \times (k+1)}{2}$ ，那么将所有的数排列在 $k \times k$ 的网格图（点 (x, y) 在网格图中当且仅当 $(1 \leq x, y \leq k)$ ）中满足 $x \leq y$ 的位置，第一次将 x 相同的归为一组，第二次将 y 相同的归为一组。那么通过每个点每次所在的块的大小就可以轻松区分出不同的块从而确定答案。

4.3 算法三

假如 n 不能表示成 $\frac{k \times (k+1)}{2}$ ，那么块大小区分法就无效了，因为怎么分都会有两个块大小相等。这时候就要想别的办法了。

还是按照上面的方法排列这些点，多出来的部分排在 $k+1$ 行的前几个位置。那么纵向的块大小已经两两不同了，只有横向有两个块相同，那么只需要区分它们就行了。

将最后一行的最后一个点拿出来，移到 $(k+1, k+1)$ ，询问的时候首先鉴定出这个点就可以区分这两个块了。

由于纵向只有 $(k+1, k+1)$ 和 (k, k) 两个点所在块大小相同，所以直接根据这两个点的横向块大小区分即可。

注意到这样做在 $n = \frac{k \times (k+1)}{2} - 1$ 时会由于横向最后两个块大小相同而无法区分。

既然 $n = \frac{k \times (k+1)}{2} - 1$ 会挂那就不拿最后一个点，随便拿一个其他的替换它就可以解决 $n = \frac{k \times (k+1)}{2} - 1$ 的情况。

但是这样做询问次数是 $2n \times \sqrt{2n} \approx 28000000$ 。所以还是不能AC。

4.4 算法四

那么还需要继续加一些常数优化。

由于每次询问之后已经被归到某一块的点显然是可以不用再次询问的，于是把已经归类的都删掉，这样就获得了一个0.4的常数。（因为随机排列后期期望选到的是更大的块，所以大块会更有可能被优先删掉，所以这个常数小于0.5）

这样就足够通过所有测试点了。

那么还能继续优化吗？

假如刚开始就能判断一个块最终大小大不大那就可以优先判断最终期望大小比较大的块减少期望次数了。

那么考虑换一种询问方法：不枚举块，而是枚举每个元素，尝试将它归入现在存在的块中，假如归不进那么就新开一块。

假如尝试的顺序就是块出现的顺序的话容易发现两种做法是等价的。但是可以不按它出现的顺序，而是可以按照当前块大小从大到小询问。

这种做法大概可以再优化掉10%左右的询问次数。