《卡常大水题》解题报告

matthew99

1 试题大意

给定一张n个点的有向完全图，边有两种权值，求两种权值的最大值之和最小的强联通子图。

2 数据范围

对于前60%的数据$n \le 50$，其中10%的数据某一个权值相同，其中20%的数据权值具有对称性即有向边$x \to y$和有向边$y \to x$的两种权值分别是相同的，对于所有数据$n \le 150$，权值在10^9范围内。

3 得分估计

估计几乎所有人都可以获得40分，一部分人可以获得60 分，一半左右的人可以获得满分或者接近满分的分数。本题属于送分题。

4 算法一

暴力枚举所有子图。复杂度$O(2^{\frac{n(n-1)}{2}})$。期望得分10分。

5 算法二

如果某一维权值为0，那么另一维二分或者直接从小到大枚举然后用tarjan判断即可。如果暴力枚举则复杂度为$O(n^4)$，二分复杂度为$O(n^2 \log n)$，期望得分10分。

6 算法三

如果权值具有对称性，那么如果有向边$x \to y$在强联通子图中，那么不妨将有向边$y \to x$也加入该子图，答案不会变，因此只要在对应的无向图上考虑即可。

无向图上强联通分量就是树，做法类似NOI2014的《魔法森林》。

结合前三个算法可以获得40分。

7 算法四

一种$O(n^4 \log n)$的做法是，枚举第一维的最大边，二分第二维的最大边，每次用tarjan判断。

一种$O(n^5)$的做法是，枚举第一维的最大边，用类似Floyd的算法求出每一对点之间最大边最小的路径中最大边的权值，那么用1号点到其他所有点和其他所有点到1号点的这个权值中的最大值减去当前最小边的边权更新答案即可。

8 算法五

我们注意到枚举最小边之后，最大边具有单调性，那么我们可以将$O(n^4 \log n)$的算法时间复杂度降为$O(n^4)$。常数只要不是太大也可以通过此题。

算法五是我估计考场上大部分AC的同学都会写的算法，但是其实我非常希望大家实现的是下面介绍的这个算法。

9 算法六

注意到如果我们从大到小枚举最小边，每一对点之间最大边最小的路径中最大边的权值实际上是可以在$O(n^2)$时间内更新的，因此时间复杂度也很容易降为$O(n^4)$。

10 算法七

算法六由于常数过大难以在时限内出解，我们考虑如何卡常数。

首先由于没有开O2因此要避免使用stl的min和max函数。

其次，我们注意到如果我们把二维数组f[i][j]压到一维数组f[$i \times n + j$]上，那么第一维上差1的位置在一维数组上差n，第二位差1 的位置在一维数组上差1，因此我们不妨手写指针来加速寻址，这样常数减少了很多，可以轻松在时限内出解。

11 偏题

我估计有一部分人会被卡常数或者看到数据范围干脆不敢写。

你问我这题是不是偏题，我可以说一句无可奉告，你又不高兴，我能怎么办？

看到数据范围为什么不敢写呢，要知道时代是在进步的，不一定算出来好几亿就不能写的，电脑速度在变快，常数可以变得更优，而且时限很宽，再者一秒能做的运算量本来就能达到几亿，所谓“一亿就会超时”这种传统观点是在常数一般的时候用的。计算几何几千万就超时了，而加减位运算逻辑运算几亿完全不是问题，要学会具体问题具体分析。

实际上常数在算法中是非常重要的，有时候卡常数也是一件技术活，比如有时候为了降低FFT的常数，需要一定的复数技巧。

本题实际上和ZJOI的小星星有着异曲同工之妙，其中一个就是本题也是存在着常数很大的复杂度相同的算法，但是标算却仅仅通过一系列的常数优化，拥有了和其他算法迥乎不同的常数。

同样本人在UOJ上出的题UR13的Yist也差不多。

实际上无论你用的是什么算法，只要足够科学最终应该都可以卡常数卡过去。~~所以本题既然大家肯定会AC的轻松愉悦，那就不要说是偏题了吧~~。

本人认为本题应该存在$o(n^4)$的算法，但是感觉应该做不到$O(n^3)$。

~~做到了就偷偷出到NOI里面然后我就滚粗了2333。~~