道路染色

coloring

【题意简述】

给出一颗 n 个节点的树，要给每一条边染一个$1\sim n-1$的颜色，染颜色 i 的代价为 i，要求同一个节点连出的所有边所染颜色都互不相同，求一个为整棵树染色的方案，使得代价之和尽量小。

$n <= 150$

【算法分析】

其实这题之所以给了较优解的分，并不是因为这题没有完美的解法，真正的原因有两个，一个是因为为了使选手得分的区分度更加明显，另一个更重要而且更邪恶的是因为标准解法很水很水，所以要为了让一些不太自信的选手以为没有完美解法= =!，不知道能不能坑到大家啊…

算法一：

从小到大依次考虑每种颜色，再依次考虑所有未染色的边，如果这条边能染这种颜色就染，这样得到一个较优的答案。

时间复杂度：$O(N^2)$

期望得分：$5\sim 15$

算法二：

重复执行算法一 T 次，每次在执行之前将所有的边打乱顺序，从 T 次得到的解答中取出最优的作为答案。当 T 的取值适当时，可以得到正确率和时间之间的平衡。

时间复杂度：$O(TN^2)$

期望得分：$35\sim 55$

算法三：

将算法一或算法二得到的解答作为初始解，进行一些盲人爬山、模拟退火等调整算法，从而得到更优的方案。

时间复杂度：N/A

期望得分：$40\sim 80$

算法四：

初看这题似乎可以考虑使用树形动态规划来解决，因为状态表示起来很方便，对于以节点 i 为根的一颗子树，他对树中其他部分的影响只有 i 与其父亲的那一条边的颜色，因此我们可以使用 $f[i][j]$表示 i 与其父亲所连边颜色为 j 的子树内最小代价。

但是当我们继续顺着这条思路想下去，似乎在转移的部分遇到了不小的困难---一个节点所有孩子连向它的边的颜色都不能相同，要保证这个，似乎普通的动态规划转移过程是无法做到的。此题的转移过程的确很不普通，其实，在转移的时候遇到的这个问题是一个十分经典的最优匹配模型。将每个孩子视为左边的节点，将每个颜色视为右边的节点，所连的边权即为在子树中所计算出的 f 值，使用 KM 算法就可以完成转移了。

我们来分析一下这个算法的时间复杂度，如果我们设节点 i 的度为 D_i，对于一个节点 i，我们在做完第一次最优匹配之后能够得到 $N-D_i$ 个 $f[i]$ 的值（所有没有被匹配的颜色 j，我们都可以求出 $f[i][j]$），之后为了求出其他的 $f[i][j]$，我们要强制令 j 不能被匹配，这样对于节点 i 总共需要做的匹配次数是 D_i 级别的，而每做一次 KM 要使用 D_i 个孩子去匹配 N 种颜色，复杂度是 $O(N^2 {D_i}^2)$ 的，因此算法的总复杂度是 $O(N^2 \sum {D_i}^3)$。

乍一眼看去，似乎这个复杂度很不靠谱，因为 N 有 150，$\sum {D_i}^3$ 也可能达到 N^3 级别，总的复杂度有可能达到 $O(N^5)$ 级别，但是我们要想到，KM 算法复杂度的水分其实是很大很大的，实际上并不太可能使其单次的复杂度达到 $O(N^2 {D_i}^2)$。对于 $N=150$,所有其他点都连向 1 号点的极限情况，这个算法也能在不到 2s 之内出解。另外，题目内事先声明了所有数据均为随机生成（单点度很大的构造卡时数据用算法 2,3 太容易骗过了，最后抉择了一下要不要出，还是放弃了），所以期望的 $\sum {D_i}^3$ 会很小很小，在理论上此题也是可以通过的。

时间复杂度：$O(N^2 \sum {D_i}^3)$

期望得分：100

【附】

coloring.cpp
C++ Source File
3 KB