

【小强的逃跑】

我发现前 3 题的代码量比较大（标程写了 7.3K），于是，这题便被修改成了一个结结实实的送分题，主要考察选手读题和实现基本算法的能力。否则的话，原本作为一个数据结构练习题，这题的标程长度很可能是前三题的和。

既然是送分题，那么就要送到家。样例解释里已经说了这题的一种算法了：每次计算的时候，根据期望的定义列方程，之后解方程（高斯消元什么的）即可。这样，一次计算的时间复杂度是 $O(N^3+M)$ ，能够拿到不少分数了。

如果连高斯消元都懒得写，那么可以进行超级暴力的 dp：设 $f[i][j]$ 表示从 i 开始走 j 步的期望。这样，取 j 为一个比较大的数，也可以算出很精确的解。

如果想写得细致一点，可以在 $O(N)$ 的时间里完成一次期望的计算：

首先，我们给每个点建立一个优先队列，来动态维护一个点的“后继”，即，连出的最小边是谁。后继关系形成一些树和一些环与树。对于带环的子图，我们可以设环上一个点开始走的期望是 x ，那么整个环的所有点的期望都可以用 x 的一次式来表示。这样，可以得到关于 x 的一个一元一次方程。进行一次除法即可解出答案。然后，就可以计算出所有会走到这个环上的点的期望了。

对于树的情况，我们设所有点的平均期望是 g ，那么所有树上的点的期望也可以写成关于 g 的一次式。再次解方程即可。

这样，就搞定了。听起来这个算法很复杂，其实写起来还算简单，只是几遍 dfs 而已。其实本题可以通过各种数据结构进行进一步的优化，不过，代码量可就有点恐怖了。