IOI2015 年国家集训队第一轮作业
解题报告

长沙市雅礼中学 杨定澄

2014 年 11 月 9 日

目录

1 Printer

1.1 题目来源

codeforces 253 E

1.2 描述

有 n 个任务，它们用 t_i, s_i, p_i 三元组表示，分别是收到时间，需要的耗时，以及优先级。保证优先级互不相同。

有一个打印机，他每个时刻都会在当前收到的未完成任务中选择优先级最高的任务实现。现在恰好有一个任务不知道优先级，但告诉了你完成这个任务的时刻。要你求出这个任务的优先级以及完成每个任务的时刻 T。

1.3 数据范围

$1 \le n \le 50000, 0 \le t_i \le 10^9, 1 \le s_i, p_i \le 10^9, 1 \le T \le 10^{15}$

1.4 关键字

二分答案；模拟；堆

1.5 做法分享

我们把问题分成以下两个部分。

1. 如果所有任务的三个参数都已知了，需要解决第二问，应该用什么方法呢？
2. 我们如何确定第一问的答案。

我们先看相对简单的第一个问题，也就是第二问。

我们考虑模拟整个工作的运作流程。

我们将工作按时间排序，接下来进行模拟。

这是个怎样的过程呢？正如题意所言，我们有一个队列；每次操作都是取出队列中优先级最高的任务；我们还要支持删除的操作。

在我制作的数据中，有 25% 的数据规模支持 $O(n^2)$ 的复杂度，相信 noip 提高组级别的选手在细心且会做第一问的情况下可以解决。

如果选手掌握了“数据结构”的知识，那么很容易意识到，我们需要用一个支持“插入”“删除最大值”“求最大值”的数据结构。

能做这三个操作的很多，对这三个比较有针对性、代表性的应该是**大根堆**。

C++ 选手可以用优先队列来代替。

考虑第一问。

之所以先思考第二问，除了问题本身更加简单以外，真正重要的一点在于，第二问的程序，本身就是第一问的一个判定问题！原问题不能解决的情况下，我们不妨思考其判定能否解决；若判定可以解决，我们只需要思考如何优化**枚举**这个过程。

如果一个问题的**判定**可以解决，我们第一个想法应该是二分答案。

我们注意到，这个问题有明显的**可二分性**，也就是说，随着优先级的增加，完成任务的时间不会减少。

如果我们使用二分答案的方法，则能对把**枚举**的次数由线性时间变为对数时间，再套上判定时间，我们很容易分析这样做的时间复杂度为 $O(n\log^2 n)$。

具体方法为，二分答案，找到最小的一个 $rank$，所谓 $rank$，表示的是他在所有任务的优先级中，排第几，并满足这种情况下任务结束时间不超过 T(由于题目保证有解，不超过 T 就会恰好等于 T)。

假设找到了这个 $rank$，他不一定是答案，因为题目要求任意任务优先级互不相同且为整数，有可能这个 $rank$ 不存在让他合法的优先级；但我们可以从他开始往后找，只要找到第一个 $rank$ 使得有他的位置即可，而不需要再判定合法性了。因为题目保证了有解，而问题又可以二分，所以**最小可行** $rank$ 之后的第一个**存在位置的** $rank$ 不合法则题目无解，矛盾。因此，不需要进行判定。

至此，已经能够解决这道题目了。

另外，通过与别人的交流，得知这题其实还有一种 $O(n\log n)$ 的做法

如何优化掉这个 log？要想优化模拟的过程是非常困难的，所以二分这一步相对好下手一些。因为二分答案是建立在单调性的基础上的，而有单调性的题，往往可以试着扫描的时候顺便维护信息从而使时间复杂度有了优化。

我们先假设特殊题目的优先级是 -1（本质上是 $-\infty$），进行一遍模拟。

经过这一步模拟，我们可以得到 $[t_x, T]$ 这个时间段里的信息：每个题目在 $[t_x, T]$ 内所使用的时间。

我们把在这个时间段里做过的题按优先级排序。

那么找到前 i 小的，使得他们的存在时间和等于 s_x，就相当于我只要让他刚好比第 i 小的大，就能让结束时间提前到恰好在 T！

用这个算法，就能够在 $O(n \log n)$ 的时间内出解。

1.6　数据制作

这题数据并不难做，随机就能覆盖大多数情况。

我们只需要先随机所有任务的信息，用标程去跑，随机一个点用得到的答案屏蔽它的 p_i 即可。

我的数据中，大概有这么几种情况。

1. 由于数据需要有梯度，我分别出了 $n = 1000$ 与 $n = 50000$ 的数据。
2. 对于参数 s_i, t_i，我让每组数据在 $[1, T]$ 内进行了随机。而 T 的大小则是在程序最开始进行随机。
3. 对于 p_i，我分两种情况。一种是 $[1, n]$ 的全排列，另一种则是保证互不相同的情况下在 $[1, 10^9]$ 里随机。

以上几种情况进行交杂，造了 20 组数据。

1.7　得分估计

此题是基础题。

对于大约 noip 水平选手，最简单的 $O(n^3)$ 的做法是应该想到的（即枚举第一问，暴力的判定）。由于是集训队作业题，并没有给出这类分数。

对于 noip 提高组选手，第一二问至少应该要有一问想得出优化方法。只要想出一种，就能把复杂度变为 $O(n^2 \log n)$，可以得到 25% 的分数。

而对于 noip 中较高水平的选手，或者说是冲击省队的选手，则应该能想出做法。

1.8　参考资料

二分查找法：http://en.wikipedia.org/wiki/Bisection_method

堆：http://en.wikipedia.org/wiki/Heap
优先队列：http://www.cplusplus.com/reference/queue/priority_queue/
详细题面：http://codeforces.com/problemset/problem/253/E

2 Pilgrims

2.1 题目来源

codeforces 348 E

2.2 描述

有一棵黑白树，你可以摧毁一个白点，一个黑点不高兴当且仅当他不能到达离他最远的黑点。要你最大化不高兴的黑点数目以及求出方案数

2.3 数据范围

$n \leq 10^5$

2.4 关键字

树；数的直径；树的中心

2.5 做法分享

根据题意就是 $O(n^2)$，不再赘述。

为了方便描述，我们先忽视所有有关黑白色的问题。

由于涉及到最远点的问题，我们引入一些中心的概念。

树中最长的路径称为树的直径。对于给定的树 T，直径不一定唯一。但可以证明各直径的中点唯一（不一定恰好是某个结点，可能在某条边的内部），我们称之为**中心**。

我们还能证明，任何点走到离他的最远点一定经过了中心。如果中心不是某个节点而在边的内部，可以想象既然经过了这个点，必然经过边上的两个点。

我在此定义概念**类中心**。大概就是说，如果中心恰好是节点，那么他也就是类中心，否则就是它所在的边的任意一个端点。（这只是我在这道题上的定义，并非学术用语）

可以证明，任何点走到离他的最远点一定经过了类中心。

既然类中心有这样优美的性质，那么它是否对我们解题有帮助呢？

我们将无根树变有根树，找到一个类中心，将它提根。

类中心的找法很容易，只需任意求出一条直径，根据定义来找即可。直径可以用做两次最远点或者树形动态规划的方法在 $O(n)$ 时间内求出来，具体算法是 noip 级别知识，不再赘述。

我们枚举每个点，我们要做的就是看有哪些黑点走到最远点必须经过他。

利用类中心的性质，我们知道任意 a 走到 a 的最远点时必须经过根。判定 a 走到 a 的最远点是否必须经过 b 分两种情况：

1. a 不在 b 子树内。我们只需记录每个子树内最深点的深度，拿 a 所在子树的深度是不是所有子树中最深、次深之类的情况出来讨论即可。

2. a 在 b 子树内。a 走到 a 的最远点，必须经过根，于是必须经过 b 了。

当然我们还要注意到题目要求的不是单纯意义上的最远点，而是黑点之间的。

我们只需稍微改变一下定义即可。我们把直径理解为黑点之间的最长路，中心、类中心的定义跟着变化；要维护的东西（如每个子树内最深点的深度）也跟着把 “点” 变成 “黑点” 即可。

问题完美解决，是 $O(n)$ 的算法，他与读入复杂度同阶，非常优秀。

稍微提一下的就是，我们把原树转化为只有黑点的树，事实上这棵树也被称之为 “虚树”。“虚树” 有很多强大的功能，不过对这题而言并没什么意义，我只是提一下有这么个东西而已。

还有就是，这题有点分治的做法。点分治就是每次找到树的重心，递归处理的一种强大的处理树上点对问题的工具。不过这道题用点分治有点大材小用，事实上还增加了复杂度。不过如果把问题改改，比如改为 “动态维护最远黑点距离” 之类的，就会变成点分治的经典问题。

2.6　数据制作

众所周知，随机一棵树，期望深度只有 $O(ln\,n)$，期望度数甚至只有 $O(1)$。如果是随机数据，那么复杂度跟高度或者度数相关的算法很容易水过去。

因此，对于和深度有关，我的数据有由一条链构成的特殊数据。

对于和度数有关，我的数据中有某个点向所有点连边的特殊数据。

对于绝大多数数据，我则是把点数大约分成三组，$\frac{n}{3}$ 的点连出一条链，其后 $\frac{n}{3}$ 的点构成菊花，剩下的 $\frac{n}{3}$ 的点则是以随机的方法构造。

绝大多数数据，m 在 $\frac{n}{3}$ 与 $\frac{2n}{3}$ 内随机；这会导致第二问的答案很小。于是，最后五组数据，我让 m 分别在 $[10, 50]$ 内与 $[50, 100]$ 内进行随机，使得第二问的答案变大。

再加上我设置了 $O(n^2)$ 可以过得 $n = 1000$ 的数据。这些特点杂糅，我做了 25 组数据。

2.7 得分估计

这题需要对 “树” 这个结构比较熟练，掌握树的中心或者树的重心之类的一些强大的工具。

noip 的选手应该能很轻松的得到暴力分。

省选、NOI 级别的选手要根据其实力来估计。总的来说应该会有大部分人想到利用中心或者重心进行解决。

2.8 参考资料

中心：参考自 noip2007 提高组题目树网的核。

重心：参考自漆子超 2009 年集训队论文。

虚树：参考自陈立杰 2013 年集训队作业与徐寅展 2014 年集训队论文。

详细题面：http://codeforces.com/problemset/problem/253/E

3 ARAM

3.1 题目来源

GCJ 2014 Final F

3.2 描述

你在玩一款游戏，每次会随机成为一个角色。有 n 种角色，你选择第 i 个角色有 p_i 的概率赢。

如果你有至少 1 游戏币，你可以选择“Reroll”，“Reroll”会消耗 1 游戏币，之后又随机成为一个角色。

你刚开始有 R 个游戏币。每玩一盘游戏，你会得到 $\frac{1}{G}$ 的游戏币。你所拥有的游戏币将以 R 为上限。

你会进行 10^{100} 次游戏，要你制定最优策略，求在此情况下期望的获胜比例。

T 组数据

3.3 数据范围

$1 \leq n \leq 1000, 1 \leq R, G \leq 20, 1 \leq T \leq 100$

原题是提交答案题，所以只要几分钟里跑得出来都是可以的

3.4 关键字

分数规划；动态规划；期望与概率；数学

3.5 做法分享

题目要求的是平均数，平均数一般直接做不好做。

我们有一个通用办法，叫做分数规划。即二分答案，假设答案是 $\frac{x}{y}$，我二分的答案是 p，若 $x - py > 0$，则 $p < \frac{x}{y}$。如此可以解决一般的和平均数有关的题。

题目说游戏进行 10^{100} 次，由于数目非常大，我们可以想象这个次数远远超过函数收敛下来使得我们的精度足够的次数。我们不妨假设题目是进行无限轮次。

还有就是所谓的最优策略。我们可以想象，假设在钱数一样的时候，抽到 i 我 “Reroll”，抽到 j 我不 “Reroll”，可是 $p_j < p_i$，这显然不是优秀的策略。于是我们知道，假设钱数一定，一定是最弱的几个会 “Reroll”，其余的则不会。

假设二分的答案为 Q，我们需要定义一个量 s，表示的是：赢一盘 +1，玩一盘 -Q，最终的期望值。

以上就是我们需要用到的前提。

设 A_i 表示钱数为 i 时，要得到 $i+\frac{1}{G}$ 的钱，过程中的 s。

而如果 $i = R$，状态的定义要改变，为再一次变回 R 过程中的期望的 s。

若 $i < 1$，则无法 “Reroll”，于是

$$A_i = \frac{1}{n}\sum_{j=1}^{n}(P_j - Q)$$

若 $1 \le i < R$，我们将角色按 p 值排序，枚举一个 k 表示最弱的 k 个要 “Reroll”

$$A_i = \frac{k}{n}\sum_{j=0}^{G} A_{i-1+j/G} + \frac{1}{n}\sum_{j=k+1}^{n}(p_j - Q)$$

移项可得 A_i 的表达式。

类似的，若 $i = R$，则

$$A_i = \frac{k}{n}\sum_{j=0}^{G-1} A_{i-1+j/G} + \frac{1}{n}\sum_{j=k+1}^{n}(p_j - Q)$$

事实上，A_R 即为所求。

为什么？

不妨设 $\forall i < R$，均有 $A_i < 0$，因为我们知道 A 单调不减，如果存在 $A_i \ge 0, i < R$，我们不如让 R 变成 i。

我们把 10^{100} 次操作这么分开看……许许多多次的从 R 变到 R，最后可能变成某个值。

而我们要算的是 s 不是比例，s 可加可减给我们带来了极大地便利。

容易知道，$A_i \ge -1$，所以最后一段的 s 的期望值是不超过 $-RG$ 的。

前面的话，如果有 T 次 R 变到 R，我们可以看作 A_RT，这是个庞大的天文数字。一个是大约 A_RT 的数，一个是大于 RG 的数，要判断的是他们的和是否大于 0，我们自然可以无视掉后面的数了。

换句话说，如果我们把 10^{100} 理解为正无穷，我们根本不必在意最后的那么一小段，而是完全可以理解为 R 到 R 到 R 无限循环下去。

于是只要拿 A_R 与 0 作比较即可，时间复杂度 $O(TnRG)$。

3.6　数据制作

这题数据制作非常简单，随机即可得到不错的效果。

数据没有什么明显的特点，基本上只有一些点数据规模小一些。

由于是提答题，所以按照原题的数据范围时限得开分钟级别。于是我没有将所有参数开到最大。

对于 n 很大的，数据组数往往很少；对于数据组数大的，数据规模往往很小。基本上数据是以这个为层次来分类的。

3.7　得分估计

从考场上 0AC 的情况以及算法的思维难度来看，这题是道很难的题。

应该绝大多数人都只会得到暴力分，个别人也许可以做出来。

3.8　参考资料

分数规划：参考自胡伯涛 2007 年集训队论文

期望：http://en.wikipedia.org/wiki/Expected_value

详细题面：https://code.google.com/codejam/contest/7214486/dashboard#s=p5