Contents

book

这道题考察找规律的能力，找出来的规律越多，分越高

把这题的模型简化一下，有一个 $1 \to n$ 的排列形成的数列，我们要用最少的操作次数把这个数列排序，每次操作都是把一个数放到整个数列的最前面。

做法一

性质一：每个数最多只会被操作一次。因为假如有一个数被往前拿了两次，显然第一次的操作是没有意义的。

做法一：暴力枚举操作的顺序，时间复杂度 $O(n! * n^2)$ ，期望得分 50

做法二

性质二：一定先操作大的数，再操作小的数。因为假如先把小的数放前面去了，再把大的数放前面去，小的数就又在大的数后面了，小的数必定还得再操作一次，然而操作两次是不划算的。

做法二：2^n 枚举要操作哪些数，这些操作按数的大小从大往小的顺序，模拟一下，然后检查一下最后的序列是否有序，复杂度 $O(n * 2^n)$，期望得分 50

做法三

性质三：假如操作了大小等于 k 的数，那么所有小于 k 的数也都得操作了。

做法三：所以我们不用 2^n 枚举，直接 m 从 1 开始从小到大枚举，表示要操作前 m 小的数，然后模拟，验证，这样复杂度为 $O(n^2)$,期望得分 80

做法四

其实 m 也是不用枚举的。

性质四：最大的数 n 是不用操作的（其他数操作好了，数 "n" 自然就在最后面了）。

于是我们先找到最大的数 "n" 的位置，从这个位置往前找，直到找到 $(n-1)$。假如找到头也没找到 $(n-1)$，那么数 "$(n-1)$" 需要操作，而一旦操作了 $(n-1)$，根据前面结论，总共就需要 $(n-1)$ 次操作了;假如找到了 $(n-1)$，那么数 "$(n-1)$"也不需要操作（和数 "n"不需要操作一个道理)。

同理，我们接着从 $(n-1)$ 的位置往前找 $(n-2)$,再从 $(n-2)$ 的位置往前找 $(n-3)\ldots$ 假如数 k 找不到了，那么就至少需要 k 次操作。这种做法的复杂度是 $O(n)$ 的，期望得分 100

score

这道题考察的是贪心思想和动态规划。

做法一

暴力枚举做题的顺序，然后按顺序在时间内能做多少算多少，时间复杂度 $O(n! * n)$ ，期望得分 35

做法二

对于 $B_i = 0$ 的情况，说明做题的顺序无所谓，则这道题可以转化为一个裸的背包dp问题：

令时间为背包容量，每道题的用时为物品的重量，每道题的得分为价值

此做法时间复杂度 $O(n * t)$ ，结合做法一后期望得分 55

做法三

对于 $B_i = 1$ 的情况，假如已经确定了要做哪些题目，显然按题目的用时从小到大做最优

那么我们先对所有题目按时间从小到大排序，然后再做一个背包dp，注意转移时加上因时间扣的分：

dp_i表示恰好用了 i 分钟的最高得分。状态转移方程为$dp_i = \max_{1\le j\le n} dp_{i-C_j} + A_j - (i * B_j)$。

最终答案是$\max_{0\le i\le t} dp_i$。

此做法时间复杂度 $O(n * t + n\log n)$ ，结合做法一后期望得分 75

做法四

还是考虑这个问题：假如已经确定了要做哪些题目，按什么顺序做这些题

目最好？

这个问题用到了类似 *NOIP2012 国王游戏* 那道题的贪心思路

假设已经确定了要做其中的 m 道题，某一个方案中做题的顺序是依次做 $x_1, x_2 \to x_m$，那么对于这个方案中任意的相邻两项 x_i,x_{i+1}，考虑交换这两项的顺序，方案是否会变得更优，交换方案中的相邻两项，只会对这两道题的得分有影响，对其余的题目不会产生影响。

如果不交换这两项，损失的分数是 $C_{x_i} * B_{x_{i+1}} + K$，如果交换这两项，损失的分数是$C_{x_{i+1}} * B_{x_i} + K$ (K 是一个常数)

所以只需要判断是否 $C_{x_i} * B_{x_{i+1}} \geq C_{x_{i+1}} * B_{x_i}$，如果此不等式成立，那么应该交换这两项。对上式移项得 $B_{x_{i+1}}/C_{x_{i+1}} > B_{x_i}/C_{x_i}$ 。所以对于一个确定的题目集合，做题的最优顺序只与每道题目的B_i/C_i有关，按每道题目扣分速度与做题时间的比值排序，按照比值从大到小做题。

因此我们先对所有的题目按照这个比值进行排序，接下来，只要按照排好的顺序，选择做哪些题目就可以了。用与做法三中的相同的dp即可。

此做法时间复杂度 $O(n * t + n \log n)$ ，期望得分 100

river

做法一

全部输出"impossible",时间复杂度 $O(1)$ ，由于出题人良心，本做法能得 5 分

做法二

暴力枚举每个木桩上放哪种圆盘，然后判断是否存在通路，时间复杂度 $O(N^{M+1} * N^2)$，期望得分 25

做法三

对于 $x_k = 0$ 的情况，即所有木桩都在一条直线上，问题转化成一个一维问题，则我们可以用一个dp来解决这部分问题：

$dp_{i,j}$ 表示覆盖到第 i个桩子的第 j 个圆盘时的最小花费，转移从小到大枚举桩子上圆盘的大小，注意加上前缀和优化

$$dp_{i,j} = \min dp_{k,l} | y_k + r_l \geq y_i - r_j + c_j$$

时间复杂度 $O(N^2 * M)$，不结合其他做法的情况下，期望得分 25 ，结合其他算法最高 85

其实这个做法比正解还难一点

做法四

将本题化为图论模型：

全图 $N * M + 2$ 个点，一个表示起点（代表 $y = 0$)，一个表示终点(代表 $y = W$)，其他的点表示每个点的每种圆盘选择，简单记为 (i, j)，表示点 i 上放圆盘 j 的点。

建边：

- (i, j) 到 (p, q) 有边当且仅当 $distance{-}between(i, p) \leq j.radius{+}q.radius$，边的权值为 $q.cost$ （第一类边）

- 起点向 (i, j) 连边，当且仅当 $i.y - coordinate \leq j.radius$ ，边权为 $j.cost$ （第二类边）

- 向终点连边，当且仅当 $i.y - cooridinate + radius >= W$,边权为 0 （第三类边）

然后从起点向终点跑最短路就行了。点数 $O(N * M)$，边数 $O(N^2 * M^2)$，期望得分 60

做法五

我们发现边的数量太大了，所以考虑减少边的数量。

首先，我们去掉一些肯定“没用”的圆盘。

对于盘子 i ,当存在盘子 j ， $R[i] \leq R[j]$ 且 $C[i] \geq C[j]$（即不仅半径小，还贵），那么盘子 i 就“没用”，去掉

去掉“没用”的盘子可以排序后用单调栈处理

我们将“有用"的盘子按半径大小升序排列（价格肯定也是升序），现在可以省掉一些第一类边：

从(i, j) 到 (p, k_1)

从(i, j) 到 (p, k_2)

假设 $k1 < k2, i \neq p$,那么我们去掉第二条边，也就是说 (i, j) 向 (p, k) 连边，只连 k 最小的那一条

取而代之的是再加上一些形如：

从 (i, j) 到 $(i, j + 1)$ 边权为 $C[j + 1] - C[j]$ 的边

这样之前去掉的从 (i, j) 到 $(p, k2)$ 的边就可以由从 (i, j) 到 $(p, k1)$ 的边，加上一系列从 $(p, k3)$ 到 $(p, k3 + 1)$ 的边替代了

也就是说 (i, j) 向 (p, k) 连边，只连 k 最小的那一条

预处理加边的复杂度 $O(N*M*N)$，点数还是 $O(N*M)$,边数 $O(N*M*N)$

最后用Dijkstra跑最短路，总复杂度为 $O(N*M*Nlog(N*M*N))$，期望得分 100 分

试题分析与总结

这套题的难度与NOIP day2高度相似，预估一等分数线 160，本班有 $x\%$ 的同学达到该分数

第一题为一道找规律的题目，仔细思考的话，半个小时之内应能想出来，80 分的做法相对好想一点。另外 50 的做法非常简单，假如实在找不到规律，推荐写 50 分做法。

第二题为一道考察dp与贪心的题目，只写做法二能得 20 分，只写做法三能得 40 分，只写做法一能得 35 分，所以将这两个做法结合一下：小数据用做法一，大数据用做法三或做法二。这种做法能骗到尽量多的分，不过对代码能力的要求较高。

第三题为一道图论题，直接爆搜就有 25 分，看出来是最短路模型，写一个暴力的最短路就有 60 分，假如dp水平高的话，可以再写一个dp，多得 25 分。假如实在没有时间，也要写一个输出"impossible"的代码， 5 分也不能随便丢

综上，$100 + 55 + 5$ 就基本够了一等线，所以在山东，只要把最简单的题目做对，剩下的题目尽量骗分就能一等。没拿到 160 分的同学就要反思一下自己的问题出在了哪里。