IOI 2015 Day 1

Yuhao Du

IIIS, Tsinghua University

January 19, 2016

Overall Statement

有一个圆环被划分成了L个区域，有N个代表队，每个队坐在一个区域里。
有N个纪念品，小Y和所有纪念品在区域0。小Y要给每队发一个纪念品，并且最后回到区域0。
然而小Y最多只能携带K个纪念品，移动到一个相邻区域要花以单位时间，问发完纪念品最短时间。
$1 \le N \le 10^7, 1 \le K \le N, 1 \le L \le 10^9$

咦，这题我怎么见过？

Problem Description

There are n apple trees planted along a cyclic road, which is L metres long. Your storehouse is built at position 0 on that cyclic road.
The ith tree is planted at position x_i, clockwise from position 0. There are a_i delicious apple(s) on the ith tree.

You only have a basket which can contain at most K apple(s). You are to start from your storehouse, pick all the apples and carry them back to your storehouse using your basket. What is your minimum distance travelled?

$1 \le n, k \le 10^5, a_i \ge 1, a_1 + a_2 + \ldots + a_n \le 10^5$
$1 \le L \le 10^9$
$0 \le x[i] \le L$

There are less than 20 huge testcases, and less than 500 small testcases.

Solution

Observation 1: 每次一定都是从0出去，然后发连续一段纪念品，然后回到0。
Proof: 如果不连续，交换两个方案答案不会变劣。
Corollary 1: 路径将这些代表队划分成若干段不交区间。
Observation 2: 考虑连续的一段为l到r，一定走0到l，0到r的最短路径。
Proof: 显然。

Solution

Observation 3: 至多一条路径绕了一圈。
Proof: 一条路径绕了一圈当且仅当起点和终点在不同的半圆内。因为区间不交，所以至多一个。
这样的路径将整个圆环划分成两部分。两部分的路径都为走到某个位置然后原路返回。

Solution

考虑前i个代表队，发完的代价为dp_i，$dp_i = dp_{i-k} + 2 * pos_i$。
因为dp_i单调，所以这样显然是对的。
反方向同理。
枚举绕一圈的路径起始位置，两边分别使用预处理的dp值计算，求最小值即可。
时间复杂度$O(N)$。

Overall Statement

这是一个交互题。
小Y有6枚互不相同的硬币，和一个天平。
天平支持4个操作：

- 询问A, B, C中最重的硬币。
- 询问A, B, C中次重的硬币。
- 询问A, B, C中最轻的硬币。
- 询问A, B, C中比D重最轻的硬币，如果不存在返回最轻的硬币。

根据按理论上最优解按谜之公式给分。

手玩时间。

Solution

6个硬币排列的熵为$\log 720$，一次得到的信息为$\log 3$。
所以至少6次。
直观理解，每次询问得到3个答案，询问k次只能区分3^k中不同的情况。

Solution

考虑决策树，每个节点有一个可行排列的集合。
一次询问之后，根据得到答案的不同分成了三个集合。
比如三个硬币的可行顺序为
$\{ABC, ACB, BAC, BCA, CAB, CBA\}$。
询问三个的最小值，根据答案分为
$\{ABC, ACB\}, \{BAC, BCA\}, \{CAB, CBA\}$。
最后叶子都只有一个元素，表示我们通过一系列的询问，得到了唯一的答案。

Solution

一个节点的决策集合为S，那么至少需要为$\log_3 |S|$步。
直观的做法，枚举决策，贪心选择尽量能均匀分的决策。
这样能得到7步的方案。

Solution

注意到$3^6 = 729$，而$6! = 720$，第i层的节点集合大小不能超过3^{6-i}，整棵树几乎是满的。
所以可以枚举每步的决策，然后爆搜。
使用这个剪枝就能很快得到一个合法的决策树。
使用这个决策树判断即可。

Overall Statement

有n个人，第i个人有个要求l_i, r_i。
现在有m个询问，每个询问有k个数$x_1, x_2, \ldots, x_k$，问能否选出一些人组成k个集合，大小分别为$x_1, x_2, \ldots, x_k$，每个集合内的人不同。其中如果第i个人在大小为x的集合中，要求$l_i \leq x \leq r_i$。
其中$n \leq 500000, m, s \leq 200000$，$s$表示所有询问的$k$总和。

内心OS: 看上去是个简单的数据结构题嘛！天朝数据结构水平世界第一！

Solution 1

按每个人的l排序，然后对于每个询问先按x排序。
然后一个显然的贪心是每次选取当前可行的人里r最小一些组成一个组。
对于单组询问，时间复杂度$O((n+k)\log n)$。

Solution 2

如果$\sum x_i > n$，那么显然无解。
把相同的x_i合并，我们只关心这样的x_i的个数。
下假设x_i两两不同并且递增。
同理可以把人也合并，如果两个人都满足
$x_i < l \le x_{i+1}, x_j \le r < x_j$，那么在这组询问下等价。
所以只要知道这样的人的个数即可，可以用一个简单的支持二维数据查询的数据结构解决。

Solution 2

经过上面这样的压缩后，可以套用Solution 1中的贪心算法解决。
注意到不同的x_i的取值不超过$O(n^{0.5})$。
那么上述算法复杂度不超过$O(k\min(k, n^{0.5})\log n)$。
总复杂度为$O(sn^{0.5}\log n)$。

Solution 2

拿到满分需要一些常数优化技巧。

- 预处理一些权值小的情况，改进二维查询的复杂度。
- k大的时候直接使用算法1。
- 使用比手写堆更快的数据结构。

Solution 3

考虑使用数据结构加速Solution 1的贪心算法。
假设将每个人按(l, r)放到二维直角坐标系上，那么一个大小为S的集合可以选取的区间就是一个矩形$0 < x \leq S, y \geq S$。
上述算法的过程为加入一个可行的矩形区域，去掉y坐标小于当前集合大小的候选点，然后删去若干个y坐标最小的点作为这个集合的点。
如果暴力维护这个每次加入的矩形，因为会有很多个矩形，所以复杂度会退化。

Solution 3

通过观察可以发现这些矩形的下边界是递减的，所以可以使用一个栈维护。如果相邻两个矩形的下边界一样，那么可以合并起来。
去掉y坐标小于当前集合大小的候选点就是简单地把栈顶的矩形拿出合并即可。
删除若干个y坐标最小的点，可以将当前矩形的下边界删到和左边的矩形下边界相同，然后合并成一个大矩形。直到删除到足够多的点为止。
这么做均摊的复杂度为线性。
这时候需要一个数据结构支持这样的操作：查询一个矩形内有多少个点和删去y坐标最小的k个点后剩下的矩形的下边界的y 坐标。这都是可持久化线段树能轻松解决的。
总的时间复杂度为$O((n+s)\log n)$。