《遇到困难睡大觉》命题报告

成都市第七中学 魏衍芃

Abstract

本文将主要介绍本人在校内训练中命制的一道传统题的解法和命题背景。

1 试题

题目描述

小 W 是个懒惰的 OIer，他正在打游戏。游戏有 n 个关卡，由于这是开放世界，他可以以任意顺序通关这个游戏。

每个关卡有一个轻松值和一个无聊值，越在后面打某个关卡，其轻松值就会越大。第 i 个关卡的初始轻松值是 a_i，初始无聊值是 b_i。这款游戏设计非常精妙，如果小 W 第 i 个通关的是 p_i 个关卡，那么他得到的轻松值是 $a_{p_i}+i\times k$，无聊值是 $b_{p_i}+i\times k$。而在整个游戏过程中，小 W 获得的轻松值是每个关卡轻松值的最小值，无聊值是每个关卡无聊值的最大值。

现在小 W 想要知道如何让自己获得的轻松值减去自己获得的无聊值最大。也即，找到一个排列 p，使得 $\min_i(a_{p_i}+i\times k)-\max_i(b_{p_i}+i\times k)$ 最大。

输入格式

第一行一个正整数 n 表示游戏的关卡数，以及一个正整数 k ，意义如题面。

之后 n 行，每行两个正整数，分别表示 b_i,a_i（**请注意这里的顺序**）

输出格式

一行一个整数表示游戏过程中轻松值减去无聊值的最大值。

样例输入

```
2 10
0 15
5 20
```

样例输出

```
10
```

数据范围

对于所有数据，满足：$1 \le a_i, b_i \le 10^9, k \times n \le 10^9, 1 \le n \le 10^5$
子任务 **1 (5 pts)**：$n \le 20$
子任务 **2 (5 pts)**：$n \le 100$
子任务 **3 (10 pts)**：$n \le 1000$
子任务 **4 (10 pts)**：$n \le 10000$
子任务 **5 (20 pts)**：$k \le 100$
子任务 **6 (10 pts)**：$n \le 50000$
子任务 **7 (40 pts)**：$n \le 100000$
时间限制：3s
空间限制：512M

2 算法介绍

2.1 观察性质

首先不难发现的是答案只会与关卡的相对位置有关。那么不妨对所有关卡的下标做一个平移（形式化地说，可以让关卡位于下标 $t, t+1, \cdots, t+n-1$，其中 t 是一个整数）。

枚举 $\max(i \times k + b_{p_i}) = M$ ，那么现在每个二元组 (a_i, b_i) 会有一个关于右端点位置的限制 $r_i = \lfloor \frac{M-b_i}{k} \rfloor$。由此同时不难发现关键的性质： **$r_i$ 的相对大小不随 M 改变**。

为了方便，以下的下标都满足都满足对于任意 M，r_i 始终单调不降（也即按照 $-b_i$ 排序二元组）。

2.2 算法一，算法二

如果将 $\max_i(b_{p_i}+i\times k)$ 所在的二元组移动到 0 下标，那么现在只会有 n 个不同的需要枚举的 M。

现在的任务是在合法的限制 r 下将二元组填满 n 个连续的下标（必须包含 0），并且要让 $\min_i(a_{p_i}+i\times k)$ 最大。

考虑确定 t（t 的定义在前文已经给出）后，如何得到一个最优的排列：

每个限制是一段后缀，所以可以考虑从后向前贪心，维护当前位置能够填的二元组。每次贪心选择 a_i 最大的二元组。（以上贪心的复杂度是 $O(n\log n)$ 的）

但是同时我们可以得到另外一个本质相同的贪心方法：将 a_i 从小到大排序，依次确定这个二元组所在的位置，从对应的 r_i 开始依次向前找第一个没有被占用的位置，将这个二元组放在此位置上。

根据以上贪心，可以得到以下结论：

Lemma 1 对于任意 M，选择最大的有解的 t 一定最优。

Proof 1 显然将 t 减少之后，所有的 r_i 不会增加 *(*会有减少是因为如果 $r_i=t+n-1$ 那么我们需要让 r_i 也跟着减少*)*。又不难发现的是第二个贪心的结果只与 r_i 有关，又因为所有的 r_i 都不会增加，所以显然不会让答案变得更优。

由霍尔定理，有 $t=\min(r_i-i+1)$，之后如果直接套用第一种贪心方案，并用堆进行维护，那么可以得到一个 $O(n^2\log n)$ 的做法，期望得分 20 分。

而第二种方案可以用并查集维护第一个没有被占用的位置，所以可以做到 $O(n^2)$，期望得分 30 分。

2.3 算法三

仔细考虑，如果我们让 M 增加 k，这个时候每个 r_i 也会增加 1，不难发现 t 也会增加 1。不难发现这个时候 $\min_i(a_{p_i}+i\times k)$ 也只会增加 k。

于是只需要考虑的是 $M \bmod k$ 的结果。

于是可以得到一个 $O(n\times k)$ 的做法，结合上述算法，期望得分 50 分。

2.4 算法四 - 第一部分

考虑二分答案 mid，并且仍然考虑枚举 M （在 $[0,k)$ 的范围内），看是否有解。

观察枚举 M 之后如何判断其是否有解：

首先类似 r_i 可以得到一个对左端点的限制 $l_i = \lceil \frac{mid-M-b_i}{k} \rceil$，类似的，$l_i$ 同样具有相对大小不变的性质。

由于已经确定了 t，那么实际上的 l_i 是 $\max(\lceil \frac{mid-M-b_i}{k} \rceil, t)$，实际上的 r_i 是 $\min(\lfloor \frac{M-b_i}{k} \rfloor, t+n-1)$。

现在可以构造一个匹配的模型，左右各 n 个点，每个左边的点连向右边的一段区间 $[l_i+t+1, r_i+t+1]$。

关于判断匹配实际上可以用霍尔定理解决，而由于每个左边的点连向右边的一个区间：

Lemma 2 只需要检查是否存在一个区间 $[L,R](L \le R)$ 满足 $l_i \ge L, r_i \le R$ 的二元组个数 $t > (R-L+1)$，就能知道是否存在完美匹配。

Proof 2 当找到一个左边的集合 s 的时候，其连向的右边节点的集合 t 如果不是区间，那么可以将 s 分成若干非空子集 $s_1, s_2, \cdots$，使得每个集合连向的右边节点的集合 t_i 都是 t 的一个极大区间。而如果 $|s| - |t| > 0$，那么一定存在一个 $|s_i| - |t_i| > 0$。

于是可以使用一个线段树解决此问题：

对于 R 做扫描线，维护所有 $[L,R]$ 的答案，遇到一个 $r_i = R$ 的二元组后将 $L \le l_i$ 的所有位置区间加 1，然后每次对前缀查询是否有 < 0 的位置即可。

只需要维护区间加以及区间的最小值即可完成这个问题。

但是这样做的复杂度有足足 $O(nk \log n \log A)$ ，其中 A 是答案的范围（在本题中是 10^9）

2.5 算法四 - 第二部分

现在考虑 M 从 0 变成 k 之后 l_i, r_i, t 的变化。

首先不难发现的是 l 和 r 的相对大小并不会改变（前已证），如果首先得出一个 l_i' 和 r_i' 以及 t'，分别表示这些值在 l_i, r_i, t 的时候的值。

不难发现：$l_i' - 1 \le l_i \le l_i'$，$r_i' \le r_i \le r_i' + 1$，并且都只会在 M 增加的过程中改变一次。以下我们分别记录其改变的时间戳（记录为：tl_i, tr_i）。

仔细观察线段树在判断时的操作，首先需要考虑的一定是某一个 l_i 或者某一个 r_i 组成的区间，并且每次操作影响到的区间是固定的。

这启发我们想办法将 k 次线段树的操作合并。

考虑一段区间 $[L,R]$，现在是要检查 $R-L+1-s(L,R)$ 的权值是否小于 0，其中 s 是区间内部的区间个数。而如果我们直接利用下标记录，那么实际上是检查 $r_j-l_i+1-s(l_i,r_j)$。

不难发现，由于 l,r 的相对大小不变，改变的值只有 r_j-l_i。

考虑 $q(i,j)=r'_j-l'_i+1-s(l_i,r_j)$，那么 $q(i,j)-1\le r_j-l_i+1-s(l_i,r_j)\le q(i,j)+1$。

现在就是查询是否存在一个 M 使得对于所有 $1\le i,j\le n$，$r_j-l_i+1-s(l_i,r_j)\ge 0$。

考虑维护 q，进行分类讨论：

1 如果 $q(i,j)<-1$，那么一定不存在合法的 k，使得这个区间 $[l_i,r_j]$ 合法。

2 如果 $q(i,j)>0$，那么所有 k 都能使得这个区间 $[l_i,r_j]$ 合法。

3 如果 $q(i,j)=0$，那么当 $k<tr_j$ 并且 $k\ge tl_i$ 的时候，这个区间不合法，否则合法。

4 如果 $q(i,j)=-1$，那么当 $k\ge tr_j$ 并且 $k<tl_i$ 的时候，这个区间合法。

考虑后两个情况，我们发现在 tr_j 确定时，取最小的 tl 能获得最紧的限制。

同时由于不能出现 <-1 的情况，如果不存在情况 1，那么 $=0$ 和 $=-1$ 的情况只能是最小和次小值。

可以考虑用类似第一部分中的线段树，每个结点维护区间最小值，次小值以及对应值下的最小 tl，对 r' 做扫描线，在每次操作后询问最小值以及次小值，得到限制。

最终会有 $O(n)$ 个限制，对于 $q(i,j)=-1$ 得到的限制，可以轻松求出它们的交（是一个区间 $[L,R]$），然后对于 $q(i,j)=0$ 的限制，可以从 L 扫到 R，同时记录覆盖的最大右端点。

于是得到了在 $O(n\log n)$ 的时间完成对 mid 的判断。

总复杂度 $O(n\log n\log A)$，其中 A 是答案的范围。

3 命题契机与过程

本题是笔者在完成集训队作业中的《Scenery》一题时，错误地转化了题意得到的结果，当时笔者完成了一个 $O(n^2 \log n)$ 的做法（也即算法一）。

而之后在校内题目的命题过程中，笔者将题意中的“判断是否大于”改为了求最大，并且得到了算法四的做法。

4 总结与展望

本题是一个数据结构题，但其中涉及了二分图匹配的知识的应用，这要求选手在完成此题时不仅对基础的数据结构技巧熟练掌握，还要对霍尔定理有一定了解。

完成本题需要从几个简单的结论出发，层层递进，不断发掘新的结论，将问题转化为二分图的判断问题并得到一个数据结构的解决方案。之后又要求选手对此数据结构的操作进行分析，从而得到在特殊情况下解决一组相似问题的方案。较长的解题步骤也要求选手拥有足够的耐心逐步分析。

5 致谢

感谢中国计算机学会提供学习和交流的平台。

感谢 OI 路上支持和帮助我的所有人。

参考文献

[(1)]a b c