Harmony 解説

原案 / Writer / 解説 : Cyanmond
Tester : Forested

問題概要

多分問題文を読むのが一番速いです。

問題文

パ研楽団には、N 人の演奏者がいます。また、パ研楽団は M 種類の楽器を所有しています。i 人目の演奏者の熟練度は A_i で担当楽器は B_i 、楽器の移行コストは C_i です。i 人目の演奏者は C_i の移行コストで担当の楽器を好きなものに変更することができます。

ある演奏の調和度を以下のように定義します。

- M 種類の楽器それぞれについて、その楽器を担当する人の熟練度の最大値を考える。この M 個の値の最小値が調和度である。ただし、担当する人がいない楽器が存在する場合、調和度は 0 である。

$x = X_1, X_2, \ldots, X_Q$ について、移行コストの総和が x を超えずに実現できる最大の調和度を求めてください。

制約 : $N, Q \leq 200000$

小課題 1

まず、$Q = 1$ の場合の解法を考える。

答えを二分探索することができないだろうか？

小課題 1

まず、$Q = 1$ の場合の解法を考える。

答えを二分探索することができないだろうか？-> できる。

コス x 以内で調和度を v 以上にできるか？という判定問題を考える。

小課題 1

コスト x 以内で調和度を v 以上にできるか？という判定問題を考える。
これは以下のように解ける。

- $A_i < v$ である要素を無視する。
- 各楽器について、元々その楽器を担当している人のうち C_i が最も大きい (移行コスト) 人はその楽器を担当することに確定。
- まだ担当がいない楽器に、移行コストが低い順に担当が決まっていない人を割り当てる。
- 全楽器に担当を割り当てられたら OK で、そうでなければ NG

この二分探索は先にソートしておけばクエリごと $O(N \log N)$ になる。
十分高速。

小課題 1

クエリが Q 個来るわけなので、$O(Q \log N)$ 通りの調和度の値について最小の移行コストを計算している。

冷静に考えて調和度の値は $O(N)$ 通りしかありえない。

それぞれについて移行コストの総和として考えられる最小値を求めておくこともできる。

先ほどと同じ要領で $O(N)$ で各調和度の値について移行コストの最小値を求められる。

小課題 1

実現する調和度が上がるにつれ、必要な移行コストが単調非減少。

各クエリでは二分探索で $O(\log N)$ で解ける。元々 X はソートされているので尺取りも OK 。

以上は前計算がボトルネックで計算量 $O(N^2)$ となる。遅い言語でも安心の速度。

小課題 3

小課題 2 はあまり本質的ではないので省略する。

満点解は、先ほどの $O(N^2)$ 解と同じ方針。

- 各調和度の値について、それを実現する移行コストの最小値を求めることを目標にする。

小課題 3

人を A を基準にソートしておく。

調和度の目標を上げながら、それを達成するためのコストを高速に求める。

まず、初期状態 (調和度の目標が 0) における各楽器の担当者 (移行コストが最も高い人) を求める。

小課題 3

人を A を基準にソートしておく。

調和度の目標を上げながら、それを達成するためのコストを高速に求める。

まず、初期状態 (調和度の目標が 0) における各楽器の担当者 (移行コストが最も高い人) を求める。

このとき、担当が決まっていない楽器の数を y とおくと、担当が決まっていない人のうち移行コスト C_i が小さい方から y 番目までの人の総和を求めたい。

小課題 3

- このとき、担当が決まっていない楽器の数を y とおくと、担当が決まっていない人のうち移行コスト C_i が小さい方から y 番目までの人の総和を求めたい。 (再掲)

これは C **基準でソートされた並びの Segment Tree** を作り、それの上で二分探索をするとよい。もしくは、クエリの性質を考えれば priority_queue などでも十分。

小課題 3

目標の調和度を上げるとき、「ある人が使えなくなる」ということが発生する。

ある人が使えなくなったとき、

- その人の担当楽器が決まっていたら (B_i が同じ人のうち C_i が最も高かったら) 次に C_i が高い人の担当楽器が固定される
  - ただしそのような人が存在しない場合、楽器 B_i を担当する人が存在しなくなる
- 担当楽器が決まっていなかったら、単にその人が使えなくなる

小課題 3

これらはいずれも std::set や Segment Tree 上の操作で表せる。うまくやると $O(N \log N)$ となる。

よって、ありうるすべての調和度に対してそれを実現するための最小コストを求めることができた。

小課題 3

これらはいずれも std::set や Segment Tree 上の操作で表せる。うまくやると $O(N \log N)$ となる。

よって、ありうるすべての調和度に対してそれを実現するための最小コストを求めることができた。

クエリにおいては、毎回二分探索をするないし尺取り法を用いるなどすればよい。計算量は、例えば尺取り法を用いると $O(N \log N + Q)$ となる。

あとがき

Segment Tree の使い方など、割と「いつもの」と言われがちな典型的な問題だったと思います。

黄色上位くらいあればかなり簡単だと感じると思います。逆に、ソートして賢く上手にやる問題に慣れていない人にはだいぶ難しかったのではないでしょうか。

おまけ

この問題は、元々あった以下の問題がほぼ既出であることから慌てて 1 日前に作られました。良ければ解いてみてください。(700 - 800 点)

- 二次元平面がある。次のクエリに答えよ。
  - クエリ 1 : x, y が与えられるので、点 (x, y) を追加する。
  - クエリ 2 : 全ての辺が x, y 軸と平行な長方形のうち、 4 頂点全てが既に追加されているものの面積の最大値を求めよ。
- $1 \leq Q \leq 100000$

統計情報

オンサイト AC - 8 人

全体 31 人

オンサイト FA : tatyam さん (20:45)

全体 FA : ei13333 さん (19:23)