



Librarian's work

原案 : tokoharu

問題文 : tubo28

問題概要

- 一列に本が N 冊並んでいて、挿入を繰り返してソートしたい。
- この挿入の時に変なコストがかかる
 - 1. 挿入される本が重さ w_i のとき、(定数 C) * (移動距離) * w_i
 - 2. 挿入の影響でずれた本の重みの和
- ソートできるまでの最小コストは？
 - $N \leq 10^5$
 - $w_i \leq 10^5$



考察

- 挿入操作は、「隣に挿入」を繰り返すことと同じ操作
 - 理由：コスト関数が線形だから
- 「隣に挿入」は「隣同士スワップ」と同じこと
- 「隣同士スワップ」では、wが小さい方を挿入することになるとコスト減
- 結局、バブルソートをし、各スワップでコストを計算すれば答えが求まる
 - 実際、転倒数を増やしても無意味。
- もちろん、愚直だと $O(N^2)$

これ見覚えがある問題ですね？

- 蟻本第二版 p.162 (バブルソートの交換回数)
 - $C=1$ だとBITに加える数を工夫すればすぐできる
- $C>1$ のときも同様にできる？
 - できる
 - $C=1$ だと思ったコストを除けば、
 - 残りは w が小さいときにだけコストがかかるので、
 - めんどくさそうなsegment treeでいける
 - 解説略
 - 時間： $O(N \log^2 N)$
 - 感想：めっちゃばぐらせた

分割統治法

- 本の順番でソート済みの列A,Bが順にある。マージするコストは？
- A_i のコストが(余分に)足しこまれる条件は？
 - (B_j の番号が A_i より番号が若い) && (B_j のほうがコスト高)
 - これはBITを用意すればいい
 - B_j が小さい順に更新クエリ
 - A_i にとって適切なタイミングで取得クエリ
- B_j のコストについても同様の方針
- $C=1$ のももちろん同時にできる
- 結局1度のマージあたり $O(N \log N)$ かかる
- 合計 $O(N \log^2 N)$

裏話・感想

- IOIの問題考えたついでに列の問題考えてたらできた
- 反転数系はJOIの人たちは鍛えられてるからパパッと書けそう
- 分割統治法はすごい
- tokoharuが仕事しない分tuboさんが頑張ってくれました

