模擬地区２０２０Ｂ

Warp Points

原案：tokoharu
問題文：not
データセット：smiken
解説：smiken
ジャッジ解：hec,hos,not,smiken,tomerun,tsutaj

問題概要

N頂点のグラフがあり初期状態では枝は存在しない.

各頂点にはポテンシャルA_iが割り当てられている.

コストC(l,r)を払うと $l \le x < y \le r$を満たす x,yの間に枝が張られる.
グラフを連結にするために必要な最小コストは？

ただし

$C(l,r) = \min_p \mathrm{sum}_{l<=x<=r} \mathrm{abs}(A_x - p)$

コストの最小値

$C(l,r) = \min_p \mathrm{sum}_{l<=x<=r}\ \mathrm{abs}(A_x - p)$

ここで区間[l,r]を決めたときコストを最小化するpは？
→ **p は A_l,…,A_r の中央値のときコスト最小**

何故？
→ コストをpの関数と見ると凸関数.
　中央値より小さい p を選ぶと，pを大きくしたほうがコスト減少
　中央値より大きい p を選ぶと，pを小さくしたほうがコスト減少

グラフを連結にするには？

隣り合う頂点同士が全て結ばれている状態になっていれば連結になる．
一方で，そうでない箇所がある場合は連結にはならない．

また，コストの定義から次のことが言える

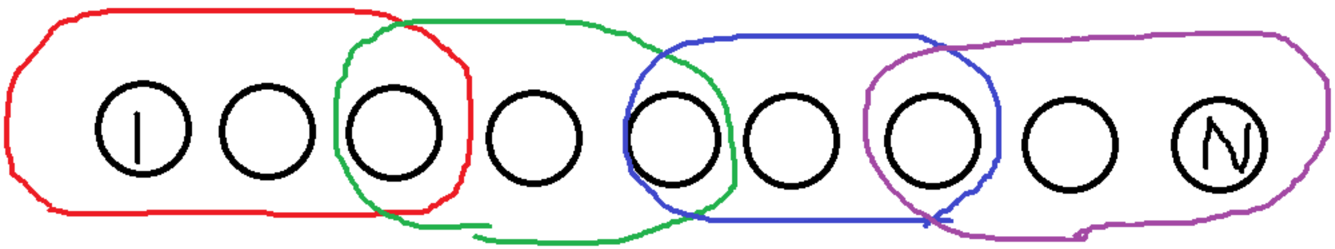
・C(l,r)は非負
・C(l,r)>=C(l+1,r) , C(l,r)>=C(l,r-1)

・ある区間が別の区間を包含している場合
→　含まれているほうの区間を消してもコストは増えない

・区間同士の積集合のサイズが2以上の場合
→　かぶっている部分のサイズを1つに減らしてもコストは増えない

最適解のかたち

区間同士の重なりは高々1箇所で，包含するものは無いので
こんなかんじになる



DP

DP[i]=頂点 i までを区間で覆った場合の最小値とする

DP[1] = 0
DP[i] = min(DP[j]+C(j,i)) (1<i<=N, 1<=j<i)

答えはDP[N]

あとはコストC(j,i)を高速に求めるのが必要

コストを高速に求める

多重集合に要素を1つずつ追加していき，そのたびにコストを求めたい

例えば以下の方法でできる.

多重集合の要素数を sz として
・小さいほうからfloor(sz/2)個の要素を保持する多重集合A
・大きいほうからceil(sz/2)個の要素を保持する多重集合B
を保持する.

要素 x 追加時には
Bの最小要素よりxが大きかったらB，そうでなければAにxを追加して，
個数が合うようにAの大きい順/Bの小さい順にもう片方の多重集合に要素を移していく.

コストを高速に求める

中央値はBの最小要素なので，Aに含まれる要素の和とBに含まれる要素の和がそれぞれ計算しておけばコストも計算できる.

解説中で「多重集合」とした部分は C++ならstd::multiset や std::priority_queueを使うと，
要素追加にO(log N)時間，コストの計算にO(1)時間で対応できるので
この問題全体ではO(N^2 log N)時間で解けます.

ちなみに O(N^2) でも解けます

統計情報

正答数　３４
提出数　６０

FA ThinkBET(会津大) 7分