

JAG ICPC模擬国内予選2026

I: 悪のロボット軍団を作ろう

原案: hint908

問題文: riantkb

データセット: potato167

解答: hos, kotatsugame, hint908, potato167

解説: potato167

問題概要

- 長さ N の数列 W と整数 K が与えられる。
- 長さ N の数列 P がすべて 0 で初期化されている。
- M 個のクエリが与えられる。
 - L_k, R_k が与えられるので、 $P[L_k], \dots, P[R_k]$ に 1 を足す。
 - その後以下の操作を好きな回数行う
 - $P[1], \dots, P[N]$ から 1 を引く。スコアが 1 増える
 - 整数 i を選んで $P[i]$ から 1 を引く
- P について、常に非負かつ $\text{sum}(P[i] * W[i]) \leq K$ を満たしていないといけない
- 適切に操作をしたときのスコアの最大値は？

解法

- 逆から考えると？
 - 以下の操作を好きな回数行う
 - $P[1], \dots, P[N]$ に 1 を足す。スコアが 1 増える
 - 整数 i を選んで $P[i]$ に 1 を足す
 - L_k, R_k が与えられるので、 $P[L_k], \dots, P[R_k]$ から 1 を引く。
 - 最終的に P がすべて 0 になるような操作のうち、スコアの最大値は？
- 「最終的に P がすべて 0」という条件をなくし、
「操作終了後、スコアから $\max(P)$ を引く」として良い
- 以降はこの設定で考える

解法

- 実は以下のようにするのが最適
 - $\text{sum}((P[i] + 1) * W[i]) \leq K$ である限り、 $P[1], \dots, P[N]$ に 1 を足す。
 - その後 $L_k \leq i \leq R_k$ かつ $P[i] = 0$ である整数 i を選んで $P[i]$ に 1 を足す
- 2 つ目の操作で $P[i]$ に 1 を足す操作で $\text{sum}(P[i] * W[i])$ が K を超えることはない
 - その直前に全体加算をしたタイミングの W と比べて、任意の要素が小さいため
- 時間計算量 $O(MN)$ で解けた

高速化

- ほしいデータ構造は？
 - 全体加算
 - $P[i] = 0$ である部分を選んで区間加算
 - 区間減算
- W の区間に対する最大値、最小値を例えば遅延セグメントツリーを用いて管理をすれば操作は行える
- 「 $P[i] = 0$ である部分を選んで区間加算」の回数が多くなりそう...
 - →実は高々 $3M$ 回しか行われない

高速化

- $1 \leq i < N$ に対する $\text{abs}(P[i] - P[i + 1])$ の総和を S とする
 - 全体加算 : S に変化なし
 - $P[i] = 0$ である部分を選んで区間加算
 - 全体加算でない限り、 S は必ず 1 以上減る
 - 全体加算は高々 M 回しか行われない
 - 区間減算 : S は高々 2 増加する
- 上記の事実を用いると $P[i] = 0$ である部分を選んで区間加算の回数が $3M$ で抑えられることがわかる
- 時間計算量 $O((N + M) \log(N))$ で解けた
- 用いるデータ構造の例
 - P 自体を管理する遅延セグメントツリー
 - $\text{abs}(P[i] - P[i + 1])$ とその prefix sum を管理するセグメントツリー

ジャッジ解

- hos CXX 436(103) lines, 11.9kB
- kotatsugame CXX 410(68) lines, 9.8kB
- hint CXX 299(90) lines, 7.3kB
- potato CXX 129(74) lines, 3.2kB

括弧の中身は自分が目視でテンプレートじゃなさそうと思った部分の行数です

統計情報

- AC teams / Trying teams
 - 3 / 4
- First Acceptance
 - yawn at the top (163 min)