

水以下コンテスト 解説

運営メンバーのなまえ

2024/03/30

X: Deque Inversion

Writer: Nichi10p

X: Deque Inversion

最初、長さ N ($2 \times N \leq 10^5$) の数列 A ($-10^5 \leq A_i \leq 10^5$) がある.

以下のクエリが Q ($1 \leq Q \leq 10^5$) 個来るので、各クエリ操作後の A の転倒数を求めよ.

- 1 x : A の末尾に x を追加する.
- 2 : A の末尾を取り除く.
- 3 x : A の先頭に x を追加する.
- 4 : A の先頭を取り除く.

ここで、操作の途中で A の長さは 2 より小さくなることはない.

X: Deque Inversion 部分点 1, 2 解法

- $N \leq 100$, $Q \leq 100$, クエリは 1, 2 のみ

X: Deque Inversion 部分点 1, 2 解法

- $N \leq 100$, $Q \leq 100$, クエリは 1, 2 のみ

末尾に追加 or 削除 クエリが飛んでくる

→ 可変長配列で管理可能！

- C++ : vector の `push_back` と `pop_back`
- Python: list の `append` と `pop`

各クエリに従って配列 A を更新後、愚直に転倒数を求めれば間に合う。

X: Deque Inversion 部分点 1, 2 解法

- $N \leq 100, Q \leq 100$

末尾に追加 or 削除 クエリが飛んでくる

→ 可変長配列で管理可能！

- C++ : vector の `push_back` と `pop_back`
- Python: list の `append` と `pop`

各クエリに従って配列 A を更新後、愚直に転倒数を求めれば間に合う。

先頭への操作が来ても ... 配列 A を新たに作り直したりすれば OK.

転倒数を求めるパートで毎回数列 A の要素の個数の 2 乗回程度計算が行われるため、全体で計算量は悪化しない。

X: Deque Inversion 部分点 3 解法

- $N \leq 2 \times 10^4$, $Q \leq 100$

X: Deque Inversion 部分点 3 解法

- $N \leq 2 \times 10^4$, $Q \leq 100$

転倒数を高速に求めたい → BIT で求められる！

転倒数とは、「"自分より左にある，自分より大きな数の個数"の総和」

X: Deque Inversion 部分点 3 解法

- $N \leq 2 \times 10^4$, $Q \leq 100$

転倒数を高速に求めたい → BIT で求められる！

転倒数とは、「"自分より左にある，自分より大きな数の個数"の総和」

$i = 1, 2, \dots, \text{len}(A)$ について，

- A_i より大きい数の個数を BIT により求める.
 - いま BIT で管理しているのは「自分より左にある数」
- BIT の A_i の値を 1 増やす.

とすれば，転倒数が求められる．

このままだと負の数が出てくるが，あらかじめ 10^6 を足しておけばよい．

X: Deque Inversion 満点解法

X: Deque Inversion 満点解法

操作により転倒数がどう変化するかを考える.

- ある数 x が A の末尾に追加されるとき
 - 転倒数は「 A のうち x より大きい数の個数」ぶん増える.
- ある数 x が A の末尾から取り除かれるとき
 - 転倒数は「 A のうち x より大きい数の個数」ぶん減る.
- ある数 x が A の先頭に追加されるとき
 - 転倒数は「 A の x より小さい数の個数」ぶん増える.
- ある数 x が A の先頭から取り除かれるとき
 - 転倒数は「 A の x より小さい数の個数」ぶん減る.

X: Deque Inversion 満点解法

操作により転倒数がどう変化するかを考える.

- ある数 x が A の末尾に追加されるとき
 - 転倒数は「 A のうち x より大きい数の個数」ぶん増える.
- ある数 x が A の末尾から取り除かれるとき
 - 転倒数は「 A のうち x より大きい数の個数」ぶん減る.
- ある数 x が A の先頭に追加されるとき
 - 転倒数は「 A の x より小さい数の個数」ぶん増える.
- ある数 x が A の先頭から取り除かれるとき
 - 転倒数は「 A の x より小さい数の個数」ぶん減る.

A の各値の個数を BIT を使って管理すれば, 高速にこの問題を AC できる!