



Segment/Interval Tree in class

by 林品諺

Sprout



知己知彼，百戰百勝

- 大家都看過影片了嗎？
- 線段樹能做哪些事情？
 - RMQ(Range Maximim/Minimum Query)
 - 區間和
- 除此之外，線段樹還有什麼應用呢？

Sprout



小故事

- 為什麼我要特別介紹普通線段樹的應用呢？
- 在我懵懂無知的童年時光，誤入歧途，學了線段樹。
- 那時每個人都對我說：線段樹可以做很多事情。
- 可是我只會RMQ跟區間和。
- 那時我常常問別人：線段樹除了算區間最大值以外還能做麼呀？
- 好幾個人都跟我說：很多呀，比如說算區間和。
- 除此之外呢？

Sprout



前情提要

- 線段樹不只是區間和與最大值
- 區間和不只是區間和
- 最大值不只是最大值
- 離線大法

Sprout



前情提要

- 線段樹不只是區間和與最大值
- 區間和不只是區間和
- 最大值不只是最大值
- 離線大法

Sprout



線段樹不只是區間和與最大值

- sprout 2016 第二次認證考 p? 「我就知道會這樣」
- 給你一個長度為 N 的序列 a ，支援以下兩種操作：
 - 1. 修改一個點的值 (單點修改)
 - 2. 對於一個區間 $[1, r]$ ，詢問找到一組 (i, j) 滿足 $1 \leq i \leq j \leq r$ ，最大化 $a[j] - a[i]$
- 原題的題敘是給你某支股票 N 天內的股價，問你在第 1 天到第 r 天之間，如果只能買入與賣出一張股票，最多可以賺多少錢。
- $N \leq 200,000$

Sprout



我不知道標題要打什麼

- 單點修改&區間詢問，這一定是線段樹！
- 如果可以把問題切成小問題，就可以用線段樹維護！
- D&C精神：不管怎樣先切成兩部份，可以merge就可以做
- 答案有三種可能：
 - 1. (i, j) 都在左半邊的
 - 2. (i, j) 都在右半邊的
 - 3. i 在左半邊，但 j 在右半邊的

Sprout



讚嘆分而治之的力量

- 斯斯有三種，答案也有三種：
 - 1. (i, j) 都在左半邊的
 - 2. (i, j) 都在右半邊的
 - 3. i 在左半邊，但 j 在右半邊的
- 1., 2.: 直接遞迴下去
- 3.: 最好的 (i, j) 是左半邊最小的 $a[i]$ 與右半邊最大的 $a[j]$
- 所以對於每個區間只要維護三個東西：
 - 區間的最大值
 - 區間的最小值
 - 區間的最佳 (i, j)
- 就解出來了 ~ ~

Sprout



這題給我們的啟示

- 只要有類似這種「可以切成小問題」的性質就可以用線段樹！
- 更精確地來說，只要merge夠快就行。

Sprout



前情提要

- 線段樹不只是區間和與最大值
- 區間和不只是區間和
- 最大值不只是最大值
- 離線大法

Sprout



區間和不只是區間和

- 有一個隊列，一開始是空的
- 有 N 個人依序過來排隊，第 i 個人會排在第 $a[i]$ 個人後面 ($a[i] == 0$ 就表示第 i 個人排在隊伍的最前面)
- 請你輸出最後的隊伍長相
- $N \leq 200,000$

Sprout



思考

- 要怎麼支援在第 k 跟 $k+1$ 人中間插入一個數字？
- 自己刻一個平衡樹！
- 可以，但是不好，太痛苦了
- 對於第 N 個人而言，他一定是在最後的隊伍的第 $a[N]$ 個人後面
- 對於第 $N-1$ 個人而言，他一定是在扣掉第 N 個人的隊伍的第 $a[N-1]$ 個人後面
- 時光倒流！！
- 只要能找到挖掉幾個空格後的第 k 個人在哪裡就好了

Sprout



所以要怎麼做？

- 時光倒流後，問題可以轉化這樣：
 - 問你當前的序列裡的第 k 個人是誰（區間詢問？）
 - 把第 k 個人拔掉（單點修改）
- 要怎麼找到做到這件事呢？
- 把一個序列的每個位子都初始化為 **1**，然後開一棵維護總和的線段樹
- 對於一個區間，如果他左半邊的**1**的數量不小於 k ，就表示第 k 個人在左半邊，反之就在右半邊
- 遞迴下去就好了！
- 修改就只是普通的維護總和的線段樹的單點修改

Sprout



前情提要

- 線段樹不只是區間和與最大值
- 區間和不只是區間和
- 最大值不只是最大值
- 離線大法

Sprout



最大值不只是最大值

- 給你一個多重集合 S ，一開始是空的
- 有三種操作：
 - 1. 把 x 加入到集合中
 - 2. 把 x 從集合裡拔掉
 - 3. 回答 S 裡面出現最多次的數字(眾數)，如果有多個就輸出最小的
- 操作數 $\leq 200,000$ ， $x \leq 200,000$

Sprout



思考

- 一個想法：開一個陣列， $a[i]$ 表示 i 在集合 S 裡面出現了幾次
- 目標，找到最大的 $a[i]$ ，有多個 $a[i]$ 時選最小的 i
- 其實有個用STL `map`就可以爽爽寫掉的作法，不過用到了一個神奇的技巧
- 可以用線段樹找到最左邊的最大值嗎...?

Sprout



數值線段樹

- 我們把「把數值當作索引值」的線段樹稱為數值線段樹
- 對 $a[i]$ 建一棵維護最大值的線段樹
- 問題就變成在線段樹上找最左邊的最大值
- 用類似「區間和不只是區間和」的方法在線段樹上走，就可以找到答案

Sprout



前情提要

- 線段樹不只是區間和與最大值
- 區間和不只是區間和
- 最大值不只是最大值
- 離線大法

Sprout



離線大法

- 給你一個長度為 N 的序列 a 以及 Q 個詢問
- 每個詢問包含一組 (l, r, x)
- 詢問一個區間 $[l, r]$ 裡有幾個比 x 大的數字

Sprout



思考

- 這東西其實可以被持久化線段樹揍掉，但我不是要講這個
- 如果只是要問區間有幾個數字比某個常數 k 大呢？
- 開一個陣列 b ， $b[i] = (a[i] \geq k ? 1 : 0)$
- 對 b 建一棵維護總和的線段樹
- 如果 x 照順序給的話有好事：直接維護 b 陣列&對應的線段樹，每個 i 只會被改到一次，複雜度會是好的
- 如果詢問的 x 一直亂跳呢？
- 離線！

Sprout



偷看後面的詢問

- 我們可以一口氣把所有的詢問讀進來，然後照 x 的大小排序
- 我們稱這種把所有詢問都讀進來之後再一起處理的技巧稱為「離線」，反之如果是看到一個詢問就回答就是「在線」
- 不過有些題目會強制在線，比如說要你回答一個詢問之後才給你下一個詢問

Sprout



區間 MEX

- 2016 IOICAMP 某場練習賽的題目
- 給你一個長度為 N 的非負整數序列與 Q 個區間詢問
- 每次問你一個區間中，沒有出現的非負整數中最小的是多少？
- Ex: 2 0 1 3 1
- $(0, 1) \rightarrow 1$
- $(1, 3) \rightarrow 2$
- $(0, 4) \rightarrow 5$
- $N, Q \leq 100,000$

Sprout



思考

- 這題其實很技巧，正常人應該都想不出來
- 這題會有種「要包到所有數字」的感覺
- 把問題想成對於每個 r ，往左走到 1 會有哪個數字沒包到
- 對於某個 r ，假設 $a[i]=a[j]=0$ 且 $i < j \leq r$ ，那麼 $a[i]$ 一定不影響所有右界為 r 的詢問的答案
- 對於每個 r 開一個陣列 $b[r][x]$ ， $b[r][x]$ 表示要包到 x 的話，1 至少要是 $b[r][x]$ （1 要比 $b[1][x]$ 小才能包到 x ）
- 對於每個 $b[r][x]$ 開一棵線段樹，對於每個 $(1, r)$ 詢問，搜尋 $b[r][x]$ 序列中，小於 1 的最左邊的 entry

Sprout



要開好多線段樹QQ

- 剛剛的做法需要開 N 個線段樹
- 不過 $b[r-1][]$ 跟 $b[r][]$ 只會差一個數字
- 持久化！
- ~~持你媽！~~ 可以離線！
- 把所有詢問對 r 排序，維護 $b[r][]$ 序列的線段樹
- 總複雜度只有 $O(N \lg N)$

Sprout