進階資料結構

PixelCat

January 17, 2025

講師簡介

- 邱翊均 PixelCat
- 2022 全國賽場外觀眾
- IOI 2023、APIO 2023 銀牌
- ICPC PCkomachi 隊員

講師簡介

- 邱翊均 PixelCat
- 2022 全國賽場外觀眾
- IOI 2023、APIO 2023 銀牌
- ICPC PCkomachi 隊員
- 因為想學習資料結構所以當資料結構講師

٩(ˊᗜˋ*)و

李超線段樹

李超線段樹

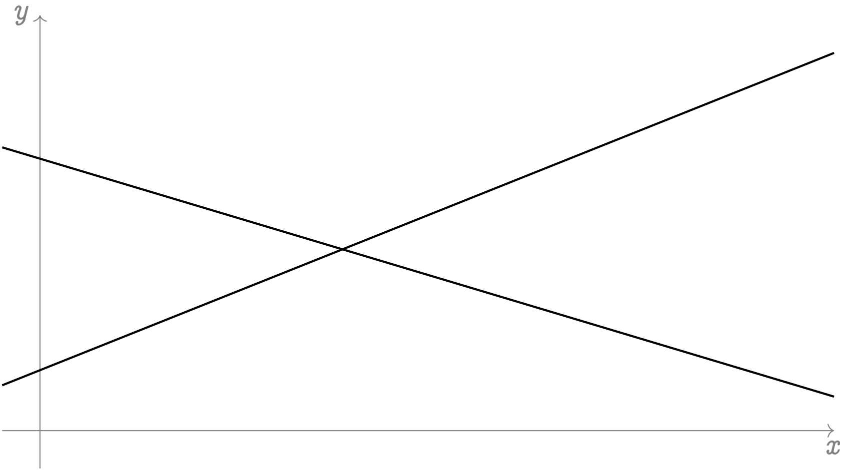
題目 (動態凸包)

現在有 Q 個操作，每個操作會是以下兩種中的一種：

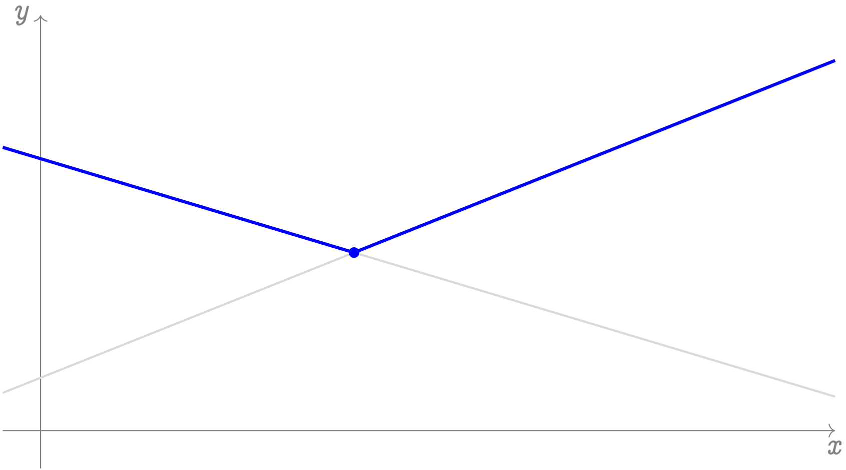
- 加入一條直線 $y = mx + k$
- 詢問在 $x = t$ 處最大的 y 值

- $1 \leq Q \leq 10^5$
- $|m|, |k| \leq 10^9$
- $1 \leq t \leq 10^5$

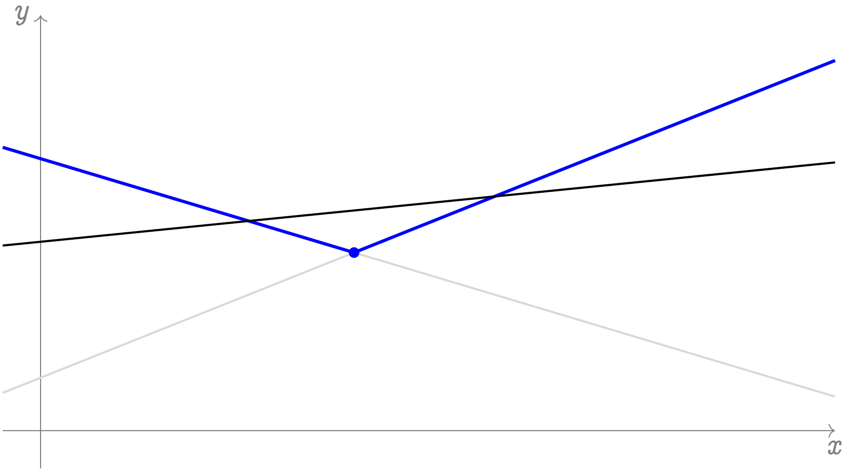
李超線段樹



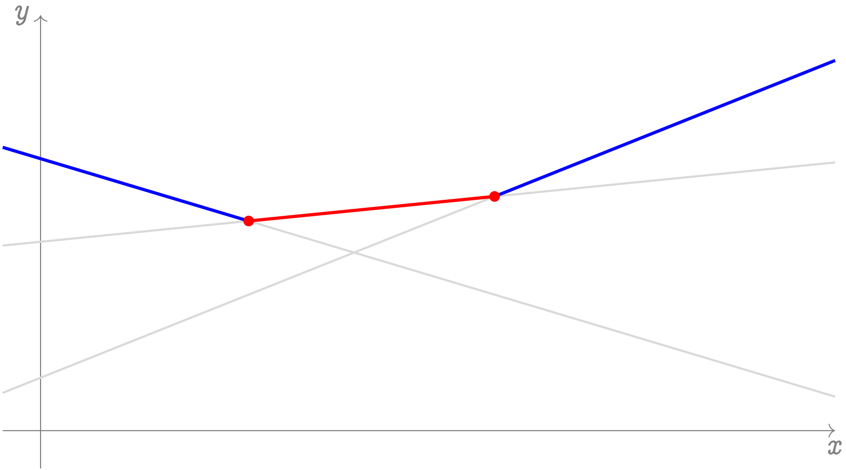
李超線段樹



李超線段樹



李超線段樹



李超線段樹

用 set 維護上凸包上的線段，維護線段控制的左右界，每次加入直線先搜他控制的區間，往左右殺掉其他線段，查詢的時候二分搜是哪條線段代值進去。注意 iterator 使用、全整求線交點……

太麻煩了，而且常數不小 ~~，而且我沒寫過~~
有沒有簡單一點的辦法？

李超線段樹

李超線段樹：

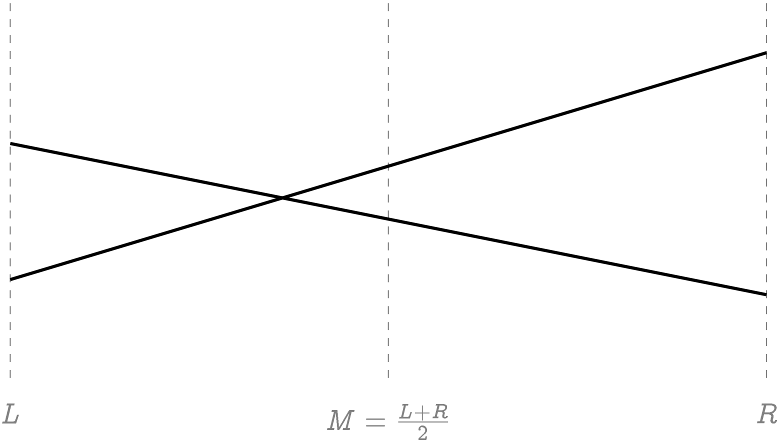
- 對要查詢的**值域**開線段樹，葉子代表單一一個 x 的值
- 每個節點存**一條**對**中點**來說 y 最大的直線
  - 對中點一定是有用的
  - 可能還對這個區間的其他一部分有用

李超線段樹 – 插入直線

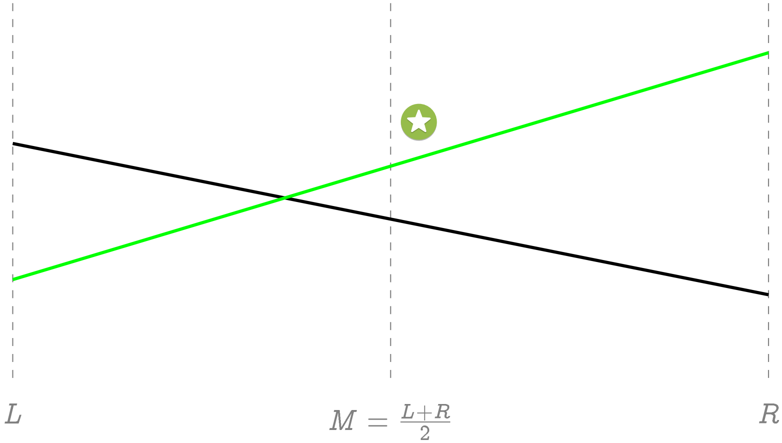
原本節點上有一條直線
這次詢問想插入另外一條直線

一個節點只能存一條線誰要留下來？另一條線要去哪裡？

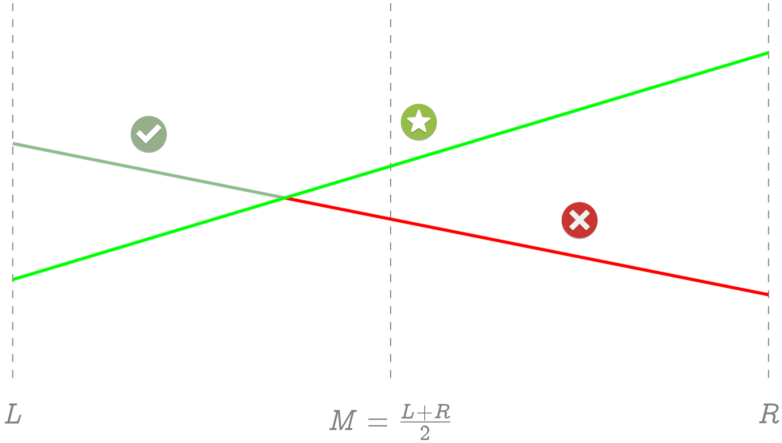
李超線段樹 – 插入直線



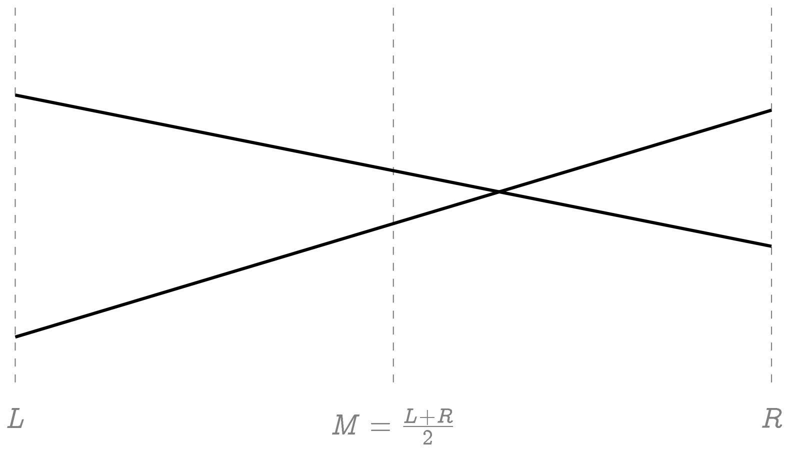
李超線段樹 – 插入直線



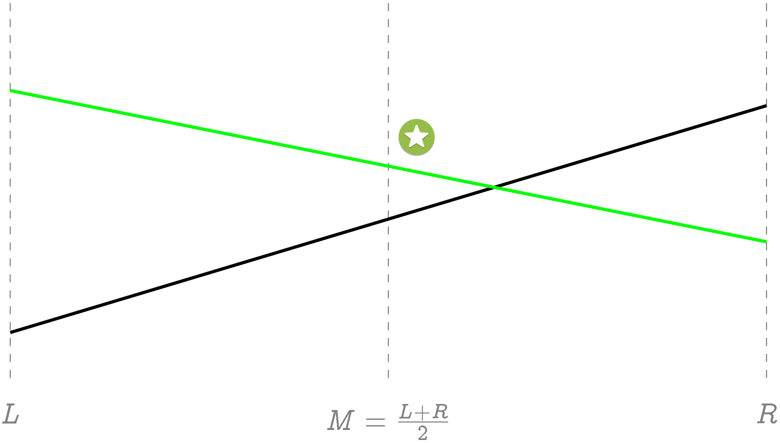
李超線段樹 – 插入直線



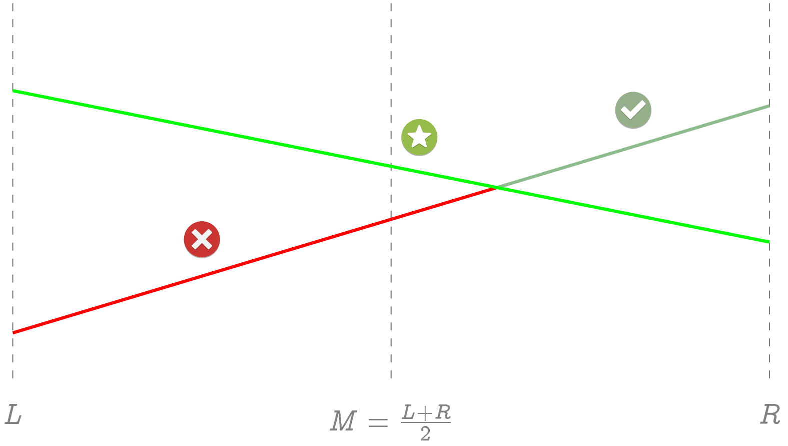
李超線段樹 – 插入直線



李超線段樹 – 插入直線



李超線段樹 – 插入直線



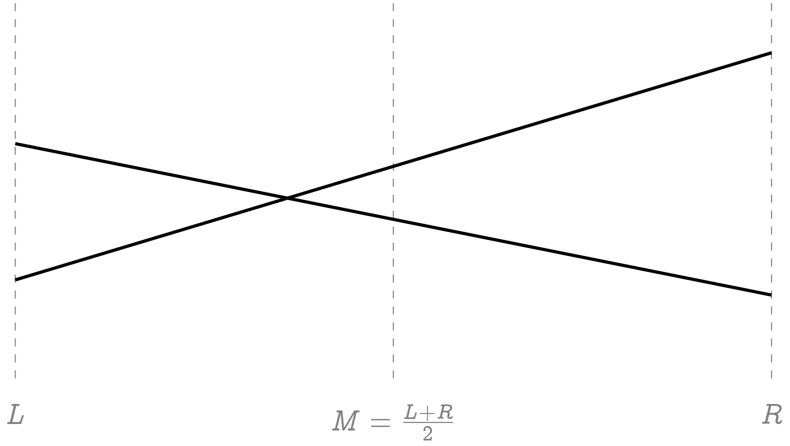
李超線段樹 – 插入直線

一條直線在中點輸掉之後不能直接扔掉，因為他還沒輸光，區間內某些 x 的範圍可能還需要他

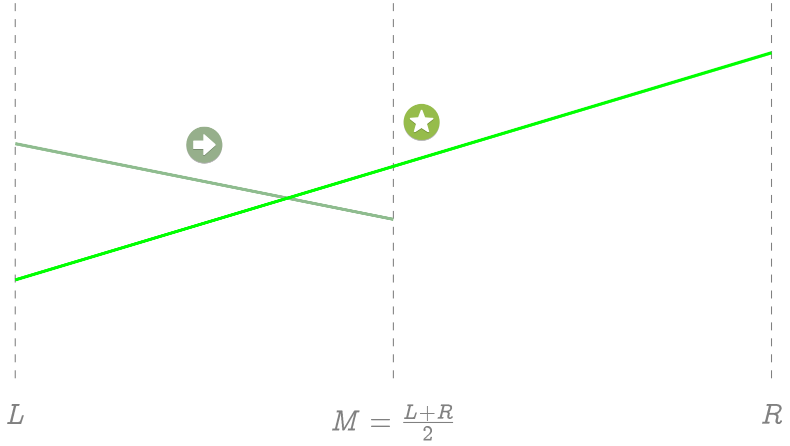
在中點輸掉的話，一定也會在左右其中一邊輸光
只有其中一邊可能還會需要用到這條直線，遞迴把他交給線段樹上那半邊的子樹處置，另外半邊已經不需要他了

到葉子還輸的話那這條線徹底不會被任何人需要

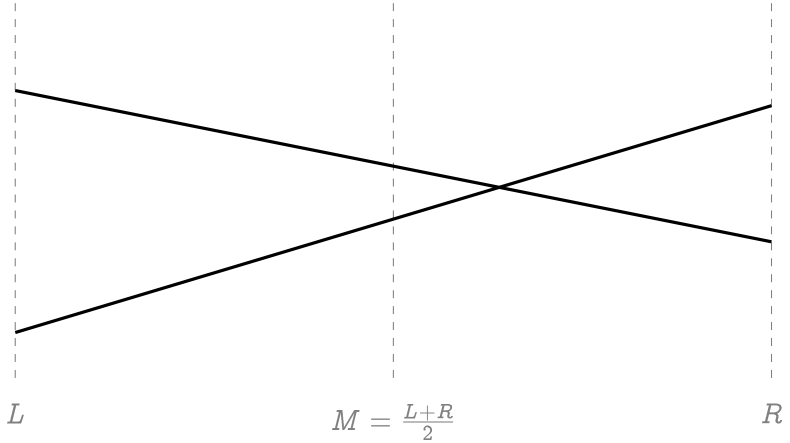
李超線段樹 – 插入直線



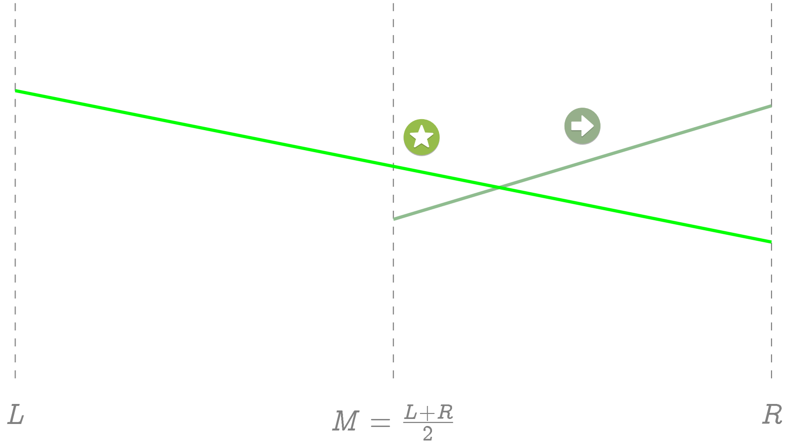
李超線段樹 – 插入直線



李超線段樹 – 插入直線



李超線段樹 – 插入直線



李超線段樹 – 插入直線

從根節點出發，到葉節點為止：

- 代中點 x 座標比較兩條直線，贏家留在節點上
- 比較兩條直線的斜率
  - 如果贏家的斜率比較**大**，輸家往**左**子樹遞迴插入
  - 如果贏家的斜率比較**小**，輸家往**右**子樹遞迴插入

一直往子樹丟包直線
時間複雜度 $O(\text{線段樹高}) = O(\log N)$

李超線段樹 – 單點查詢

直線被扔到隔壁節點，代表這個範圍的 x 全都用不到這條直線
一個 x 可能用到的直線，都存在他的祖先們身上

- 找到代表這個 x 值的葉子
- 檢查所有祖先存的直線，每個都代一次，回答最大的 y

時間複雜度 $O(\text{線段樹高}) = O(\log N)$

李超線段樹 – 實做

包裝直線作為函數使用

```
struct Line {
    int a, b;  // y = ax + b
    Line(int _a = 0, int _b = 0): a(_a), b(_b) {}
    int operator()(int x) { return a * x + b; }
};
```

李超線段樹 – 實做

插入直線

- 代中點 x 座標比較兩條直線，贏家留在節點上
- 比較兩條直線的斜率
- 遞迴插入

```
void insert(int id, int l, int r, Line ln) {
    int m = (l + r) / 2;
    if(lns[id](m) < ln(m)) swap(lns[id], ln);
    if(l == r) return;
    if(lns[id].a > ln.a) insert(L(id), l, m, ln);
    else insert(R(id), m + 1, r, ln);
}
```

李超線段樹 – 實做

單點查詢

- 找到代表這個 x 值的葉子
- 檢查所有祖先存的直線，每個都代一次，回答最大的 y

```
int qry(int id, int l, int r, int x) {
    int m = (l + r) / 2;
    int res = lns[id](x);
    if(l == r) return res;
    if(x <= m) res = max(res, qry(L(id), l, m, x));
    else res = max(res, qry(R(id), m + 1, r, x));
    return res;
}
```

李超線段樹 – 座標壓縮

題目 (Line Add Get Min，Library Checker)

你有 N 條直線 $y = a_i x + b_i$。請你處理 Q 個詢問：

- 加入一條直線 $y = ax + b$
- 詢問 $x = p$ 處最小的 y 值

- $1 \leq N, Q \leq 2 \times 10^5$
- $|a_i|, |p| \leq 10^9$
- $|b_i| \leq 10^{18}$

李超線段樹 – 座標壓縮

剛剛對 x 的值域 $1, 2, \ldots, 10^5$ 開線段樹

現在事先收集會被詢問到的 x 座標
對會被問到的 x 開線段樹

詢問是浮點數的時候也可以

李超線段樹 – 座標壓縮

如果事先不知道詢問位置呢？

李超線段樹 – 座標壓縮

如果事先不知道詢問位置呢？

動態開點，用不到的節點不要理他

李超線段樹 – 插入線段

如果插入的不是直線，而是有左右範圍限制的線段呢？

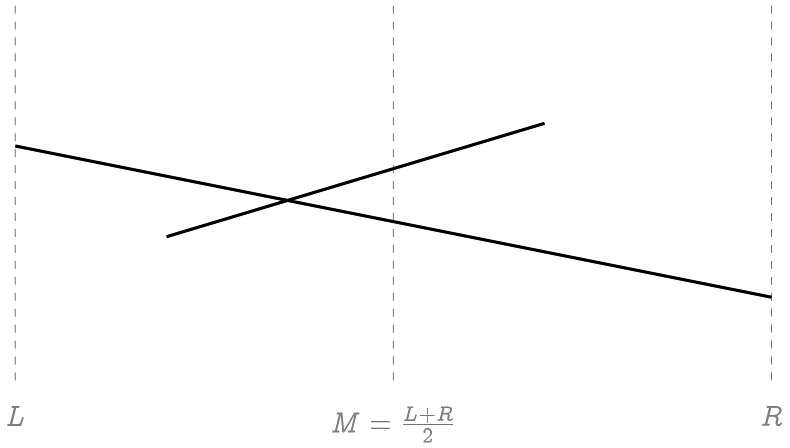
題目 (Segment Add Get Min，Library Checker)

你有 N 段**線段** $y = a_i x + b_i$ $(x \in [l_i, r_i))$。請你處理 Q 個詢問：

- 加入一段**線段** $y = ax + b$ $(x \in [l, r))$
- 詢問 $x = p$ 處最小的 y 值

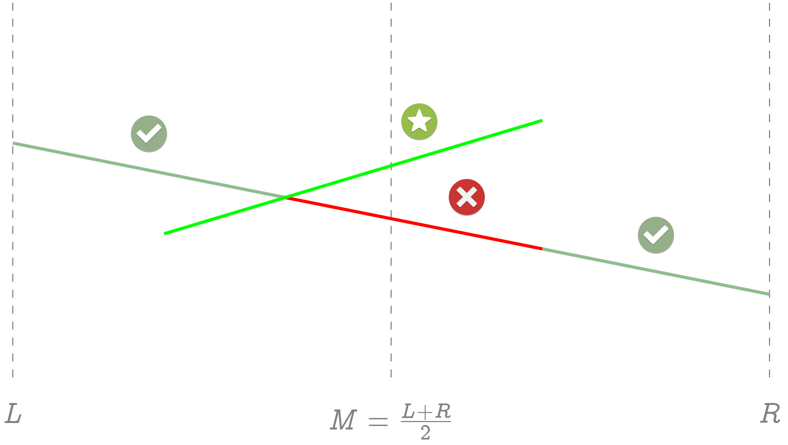
- $1 \le N, Q \le 2 \times 10^5$
- $-10^9 \le l_i < r_i \le 10^9$
- $|a_i|, |p| \le 10^9$
- $|b_i| \le 10^{18}$

李超線段樹 – 插入線段



李超線段樹 – 插入線段

(往左右子樹都丟包線段一定是不行的)



李超線段樹 – 插入線段

一般線段樹是怎麼做區間修改的？

李超線段樹 – 插入線段

一般線段樹是怎麼做區間修改的？
找 $O(\log N)$ 個節點覆蓋詢問的區間，修改那些節點

李超線段樹 – 插入線段

一般線段樹是怎麼做區間修改的？
找 $O(\log N)$ 個節點覆蓋詢問的區間，修改那些節點

找 $O(\log N)$ 個節點覆蓋線段範圍，對那些節點插入直線

李超線段樹 – 插入線段

一般線段樹是怎麼做區間修改的？
找 $O(\log N)$ 個節點覆蓋詢問的區間，修改那些節點

找 $O(\log N)$ 個節點覆蓋線段範圍，對那些節點插入直線

時間複雜度：插入一次 $O(\log N)$，總共 $O(\log^2 N)$

李超線段樹 – 應用

- 斜率優化 ✅

李超線段樹 – 應用

- 斜率優化 ✅
  - 詢問、斜率單調？單調 queue、stack？
  - 真的每次都有必要用到李超嗎？

李超線段樹 – 應用

- 斜率優化 ✅
  - 詢問、斜率單調？單調 queue、stack？
  - 真的每次都有必要用到李超嗎？
- 四邊形優化 ✅

不只是直線，有**優超性**的函數都可以

李超線段樹 – 擴充版本

一般李超線段樹

- 區間跟直線取 max（$O(\log^2 N)$）
- 單點求值（$O(\log N)$）

李超線段樹 – 擴充版本

擴充（一）

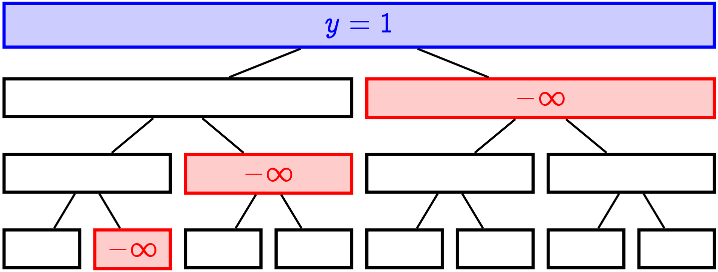
- 區間跟直線取 max（$O(\log^2 N)$）
- **區間加直線**（$O(\log^2 N)$）
- 單點求值（$O(\log N)$）

擴充（二）

- 區間跟直線取 max（$O(\log^2 N)$）
- **區間加值**（$O(\log^2 N)$）
- **區間求最大值**（$O(\log N)$）

李超線段樹 – 擴充版本（一）

前面的作法照做？加直線的標記可能會跑到以前插入的直線底下



講義範例

李超線段樹 – 擴充版本（一）

差別在於，一般李超線段樹的操作**可以交換**
有區間加直線之後操作**不能交換**

李超線段樹 – 擴充版本（一）

差別在於，一般李超線段樹的操作**可以交換**
有區間加直線之後操作**不能交換**

在一般線段樹上，我們用**下推標記**來解決標記的順序問題
同理，李超線段樹上的直線標記也需要往下推

李超線段樹 – 擴充版本（一）

下推直線：對左右子樹都做一次插入直線，清掉原本紀錄的直線
加值時，往下遞迴之前先下推當前節點上的直線

下推一次時間複雜度 $O(\log N)$
區間加直線的時候要下推 $O(\log N)$ 次
總複雜度 $O(\log^2 N)$

李超線段樹 – 擴充版本（二）

同樣的道理，加值時，往下遞迴之前先下推當前節點上的直線區間最大值像一般線段樹一樣維護

李超線段樹 – 擴充版本

李超線段樹上，節點上的直線也可以視為一種懶人標記！

李超線段樹 – 擴充版本

那，為什麼

- 擴充（一）
  - 區間跟直線取 max（$O(\log^2 N)$）
  - **區間加直線**（$O(\log^2 N)$）
  - 單點求值（$O(\log N)$）
- 擴充（二）
  - 區間跟直線取 max（$O(\log^2 N)$）
  - **區間加值**（$O(\log^2 N)$）
  - **區間求最大值**（$O(\log N)$）

不能是

- 擴充（一）＋（二）
  - 區間跟直線取 max（$O(\log^2 N)$）
  - **區間加直線**（$O(\log^2 N)$）
  - **區間求最大值**（$O(\log N)$）

李超線段樹 – 擴充版本

- 擴充（一）+（二）
  - 區間跟直線取 max（$O(\log^2 N)$）
  - **區間加直線**（$O(\log^2 N)$）
  - **區間求最大值**（$O(\log N)$）

講義有討論，不過還是值得給大家思考

時間線段樹

時間線段樹

時間線段樹想解決「操作有時效性」的問題

你想要在未來的某個時間取消這次操作
但資料結構只有支援你做新的操作、和取消**上一次**操作

時間線段樹

例如：

- 你有一個 stack，想要加新元素、和移除任意久以前加進去的東西。但是 stack 只支援移除**上一次**加進去的元素
- 你有一個併查集，想要加新的邊、和移除任意一條加過的邊。但是併查集只支援刪除**上一次**加進去的邊

時間線段樹

時間線段樹想解決「操作有時效性」的問題

你想要在未來的某個時間取消這次操作
但資料結構只有支援你做新的操作、和取消**上一次**操作

如果**操作沒有順序性**的話
可以以複雜度乘 $O(\log Q)$ 的代價，使用時間線段樹支援 Q 次這類操作

時間線段樹

線段樹的實做框架（講義第 79 頁）

- 先收集每個操作的有效時間段，加到時間線段樹裡
- 遍歷時間線段樹重現所有操作

時間線段樹

題目（【Gate】這個笑容由我來守護 - EXTREME，TIOJ 1903)

給定一張 N 個點的無向圖，一開始圖上有 M 條邊。
現在有 Q 個操作，每個操作會是以下兩種中的一種：

- 增加一條連接著編號 A_i 與編號 B_i 的邊。
- 刪除一條連接著編號 A_i 與編號 B_i 的邊（保證這條邊是存在的）。

每次操作完後請輸出當前的連通塊數量。

- $1 \leq N \leq 5 \times 10^5$。
- $Q \leq 5 \times 10^5$。

時間線段樹

- 時間線段樹對「操作時間」開一棵線段樹
- 每個節點存一些操作
- 事先收集所有操作，在線段樹上的一些節點存起來
- `traversal` 遍歷線段樹，重現所有操作

時間線段樹

如果開始到結束經歷 Q 次加邊或刪邊
對這 Q 個時間點開線段樹
每個葉子的所有祖先存的所有邊，會是這個時間點應該要活著的所有邊

時間線段樹

加入操作：像區間修改一樣，把要加的邊紀錄在被修改到的節點上

```
void insert_event(int idx, int lb, int rb, int ql, int qr, Eve
    if (ql <= lb && rb <= qr) {
        tr[idx].push_back(e);
        return;
    }
    int mid = (lb + rb) / 2;
    if(ql <= mid)
        insert_event(idx * 2, lb, mid, ql, qr, e);
    if(qr > mid)
        insert_event(idx * 2 + 1, mid + 1, rb, ql, qr, e);
}
```

時間線段樹

遍歷線段樹：進入節點時把剛剛加的邊加進 DSU，離開節點時移除這些邊

```
void traversal(int idx, int lb, int rb) {
    int cnt = 0;
    for (auto e : tr[idx])
        if (do_things(e))
            cnt++;
    if (lb == rb) {
        // 記錄在這個時間點的答案
    } else {
        int mid = (lb + rb) / 2;
        traversal(idx * 2, lb, mid);
        traversal(idx * 2 + 1, mid + 1, rb);
    }
    while (cnt--) undo();
}
```

時間線段樹

- DFS 的時候每個節點在最後剛好把自己加進去的邊拿掉
- 葉子代表一個時間點，每次 DFS 到葉子的時候，DSU 裡面正好有所有應該活著的邊

時間線段樹

最後一個技術困難：DSU 要怎麼取消上一次操作？

- 每次 DSU 加邊只會把一個節點接到另一個身上
- 紀錄每次加邊是誰連到誰，undo 的時候還原這兩個節點的狀態
- 啟發式合併？✅
- 路徑壓縮？❌

時間線段樹

Recap

- 有一個可以 undo 的併查集
- 收集每一條邊存活的時間，加入時間線段樹
- 遍歷時間線段樹，一邊紀錄答案

時間線段樹

時間複雜度

- 總共出現過 $O(M+Q)$ 條邊
- 每條邊在線段樹上被拆成 $O(\log Q)$ 個操作
- 每個操作要在併查集上 $O(\log N)$ 加邊、$O(1)$ undo

總複雜度 $O((M+Q)\log Q \log N)$
比只有加邊的啟發式合併併查集多一個 $\log$

時間線段樹

使用時間線段樹的場合

- 操作有時效性
- 操作可以交換
- 可以反悔上一個操作

線段樹優化建圖

線段樹優化建圖

題目 (Legacy，Codeforces 786B)

給定一張 N 個點的有向圖，接下來有 Q 次加邊的操作，每次操作會是以下三種中的一種：

- $1\ v\ u\ w$：從 v 到 u 建一條權重為 w 的邊。
- $2\ v\ l\ r\ w$：從 v 到 $[l, r]$ 區間內所有點分別都建一條權重為 w 的邊。
- $3\ v\ l\ r\ w$：從 $[l, r]$ 區間內所有點到 v 分別都建一條權重為 w 的邊。

請你輸出給定的源點 s 到所有點的最短路徑長。

- $1 \leq N, Q \leq 10^5$。

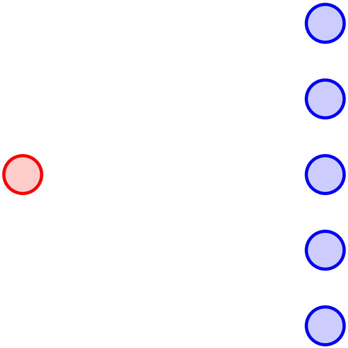
線段樹優化建圖

怎麼看起來跟線段樹沒什麼關係

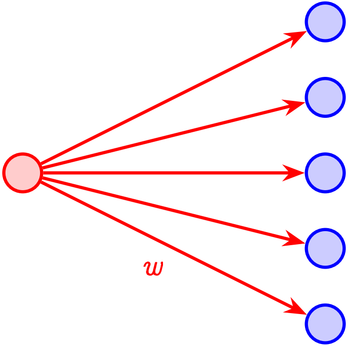
不急著砸是好事，我們先遺忘世界上所有資料結構

線段樹優化建圖

如果每次詢問都是「一個點對所有點，分別都建一條權重為 w 的邊」要怎麼辦？

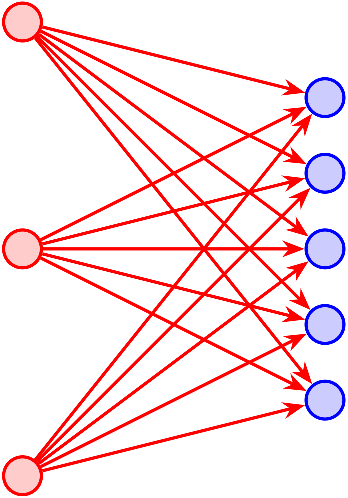


線段樹優化建圖 – 「代理人」



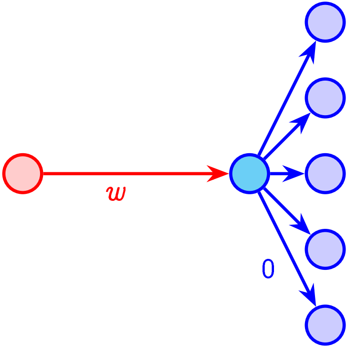
注意到，建出來每一條邊都長得一模一樣

線段樹優化建圖 – 「代理人」



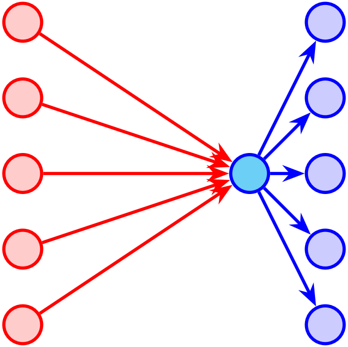
線段樹優化建圖 – 「代理人」

一次詢問建出太多邊了



先建一個中間點，中間點再連藍色點
詢問的時候，紅色點連**一條邊**到中間點

線段樹優化建圖 – 「代理人」



線段樹優化建圖 – 「代理人」

以鬆弛的角度來說，連 (a, b) 邊權 w，造成 $d(a) + w \geq d(b)$

a 連中間人 c，$d(a) + w \geq d(c)$
中間人 c 連 b_i，$d(c) + 0 \geq d(b_i)$
$\Longrightarrow$ 實質上等同 a 連 b_i，$d(a) + w \geq d(b_i)$

線段樹優化建圖

回到原本的問題，每次詢問要連邊的區間不一樣，每次開新的中間點的話問題沒有半點解決

如果可以預先決定少少的中間點，每個中間點連到一個區間？
如果可以預先決定一些區間，讓每個詢問都可以被這些區間拆分成少少段？

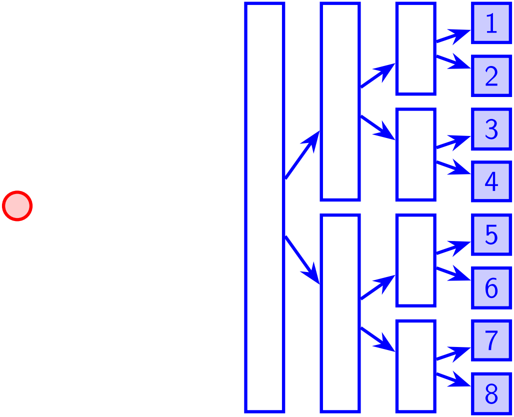
線段樹優化建圖

回到原本的問題，每次詢問要連邊的區間不一樣，每次開新的中間點的話問題沒有半點解決

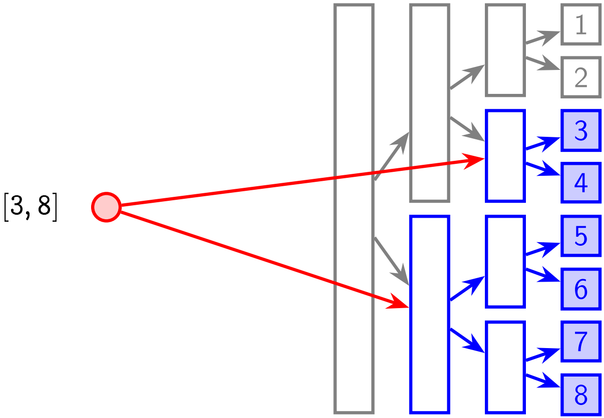
如果可以預先決定少少的中間點，每個中間點連到一個區間？
如果可以預先決定一些區間，讓每個詢問都可以被這些區間拆分成少少段？

把中間點開成**線段樹**的樣子

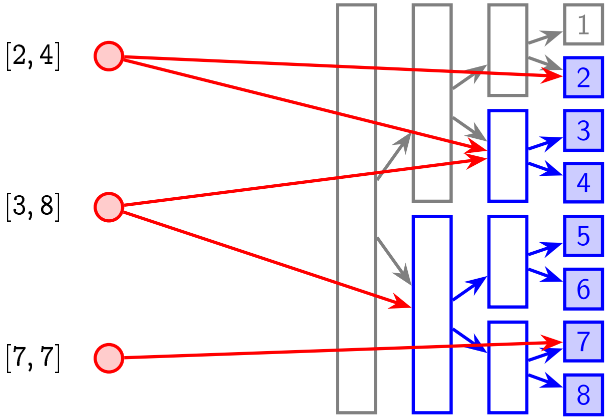
線段樹優化建圖



線段樹優化建圖



線段樹優化建圖

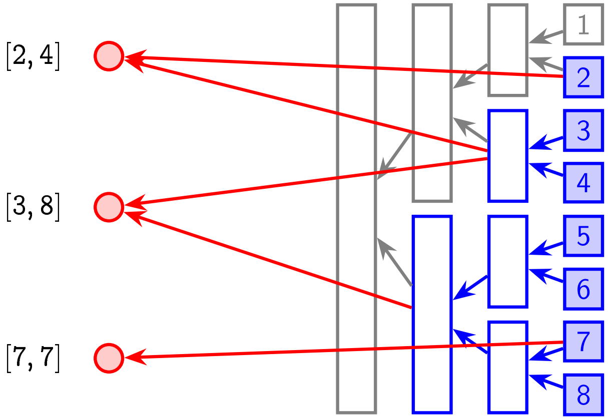


線段樹優化建圖

對一個線段樹節點連一條邊，等同於對區間內所有點分別連邊

如果把整張圖反過來：從線段樹節點連出來，等同於從區間內所有點分別連出來

線段樹優化建圖

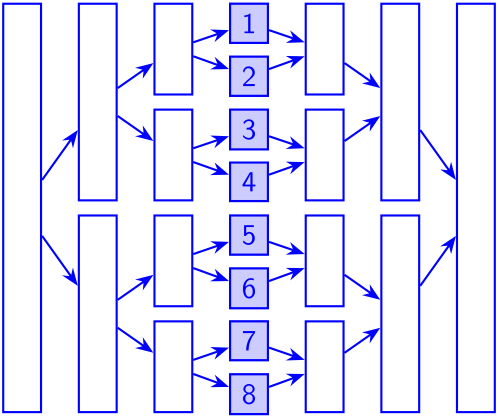


線段樹優化建圖

預先建兩棵線段樹，一棵從根往葉子連邊，一棵從葉子往根連邊
每次詢問根據方向在對應的樹上連 $O(\log N)$ 條邊，就可以建出一樣的圖（在最短路的意義上一樣）

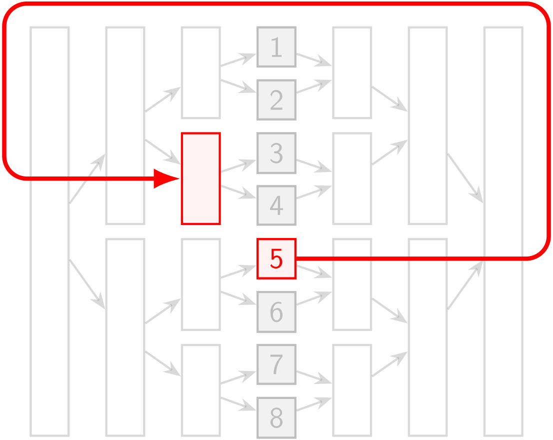
單源最短路？dijkstra 就好

線段樹優化建圖



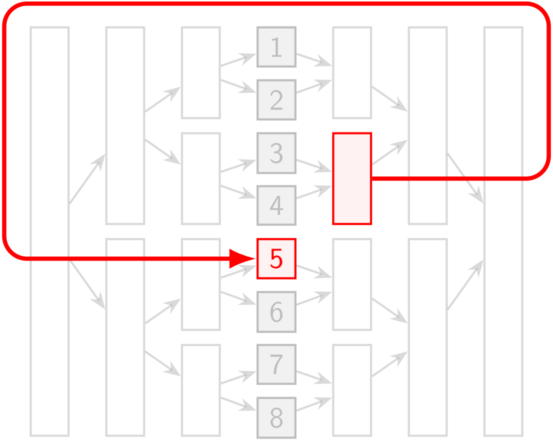
線段樹優化建圖

節點 5 連到區間 $[3, 4]$



線段樹優化建圖

區間 $[3,4]$ 連到節點 5



線段樹優化建圖

最後建出來的圖上：

- 一棵線段樹有 $2N$ 個節點，但是兩棵線段樹葉節點可以共用，總共 $3N$ 個點
- 兩棵線段樹各建 $O(N)$ 條邊，之後每個詢問建 $O(\log N)$ 條邊，總共 $O(N + Q \log N)$ 條邊

時間複雜度 $O((N + Q \log N) \log N)$

線段樹優化建圖

線段樹本身和我們是怎麼建圖的並**沒有**關係
我們甚至可以用 sparse table 建類似的圖

線段樹在這裡發揮的最大價值是**把詢問區間拆解**成一些特別的小區間

Pattern

Pattern

資料結構往往不會赤裸出現

不是「我要用這個資結砸掉這題」
而是「這題需要這樣做，所以可以拿這個資結砸掉」
永遠都先想怎麼做，再尋找合適的資結幫助你

– 2024 基礎資結

Pattern

> *資料結構往往不會赤裸出現*
>
> *不是「我要用這個資結砸掉這題」*
> *而是「這題需要這樣做，所以可以拿這個資結砸掉」*
> *永遠都先想怎麼做，再尋找合適的資結幫助你*
>
> *– 2024 基礎資結*

有時候，題目要你維護的東西實在是太荒謬了，你需要自己創造可以維護的東西去維護

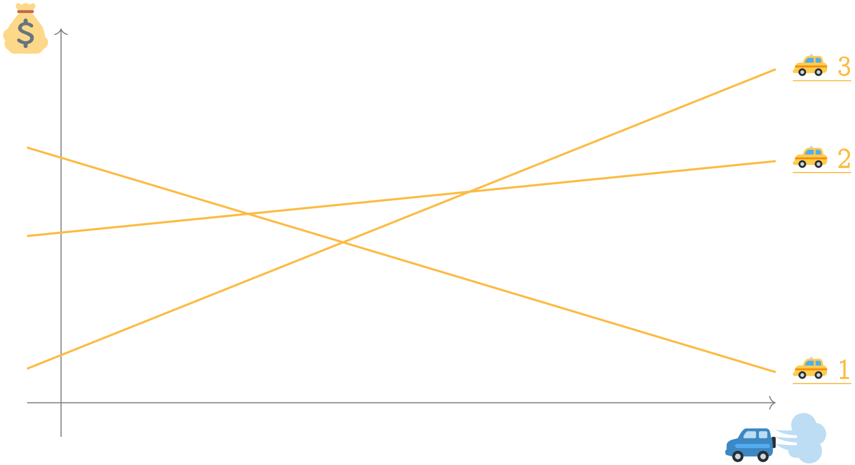
Pattern – Taxis

題目 (Taxis，POI 2018)

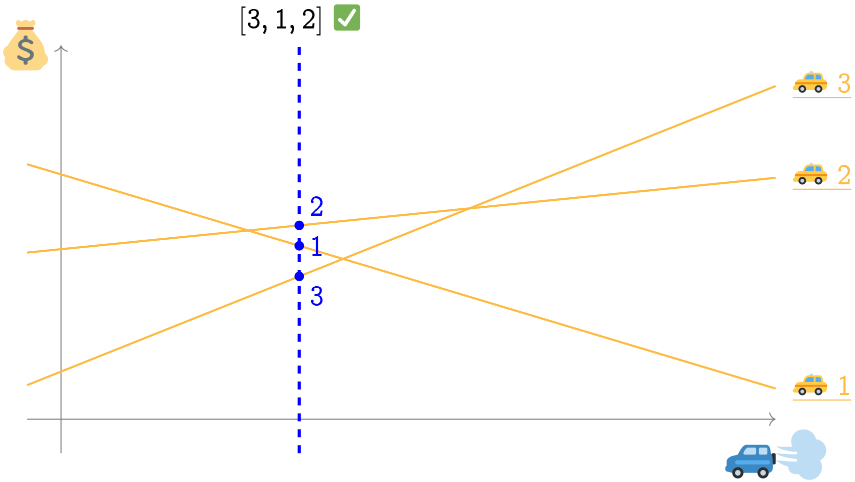
現在有 n 台計程車編號 $1 \sim n$，對於第 i 台計程車，給定 s_i, c_i，代表該台計程車的收費方式為 $s_i + d \times c_i$，其中 d 為里程數。
給定一個 $1 \sim n$ 的排列，請問是否存在一個里程數 x/y，使得把所有計程車的編號照著收費由小到大列出恰好是這個排列（當收費價格一樣，順序可以任意安排），若存在的話請輸出任意一組解，否則輸出「NIE」。
，每筆修改會有 a_i, b_i，代表交換排列在 a_i 和 b_i 的數字，每次交換後皆須輸出先前問題的答案。

- $1 \leq n \leq 5 \times 10^5$。
- $1 \leq q \leq 5 \times 10^5$。

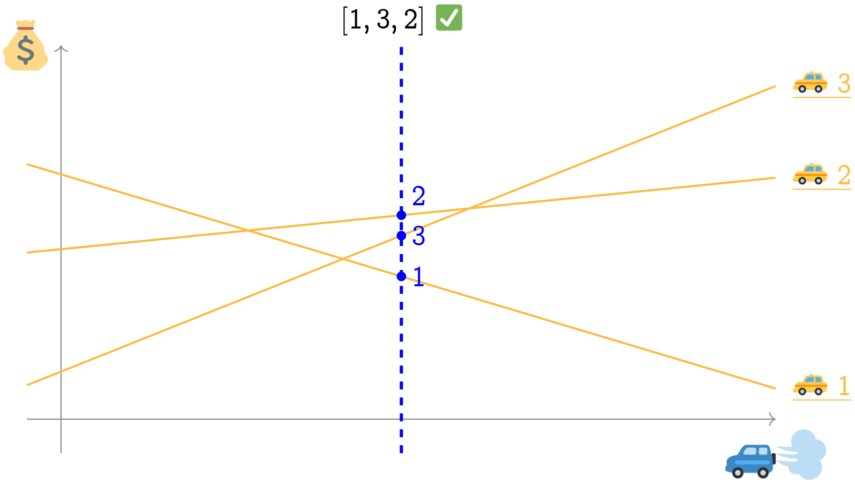
Pattern – Taxis



Pattern – Taxis

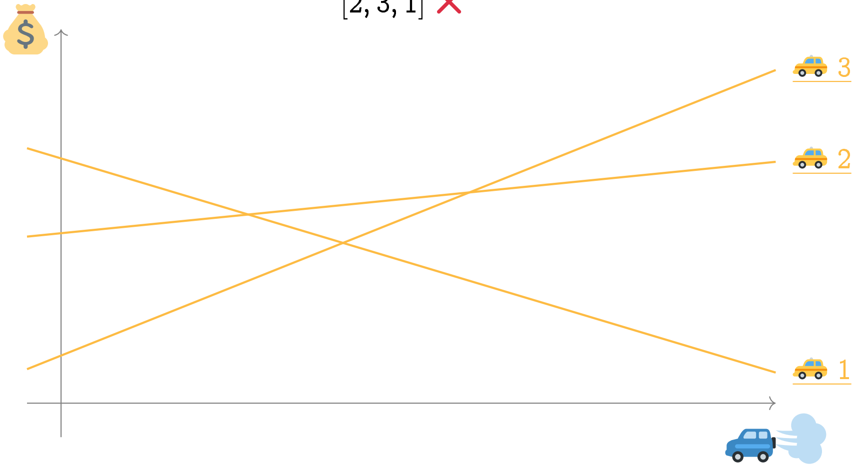


Pattern – Taxis



Pattern – Taxis

$[2, 3, 1]$ ✗



Pattern – Taxis

題目 (Taxis，POI 2018)

現在有 n 台計程車編號 $1 \sim n$，對於第 i 台計程車，給定 s_i, c_i，代表該台計程車的收費方式為 $s_i + d \times c_i$，其中 d 為里程數。
給定一個 $1 \sim n$ 的排列，請問是否存在一個里程數 x/y，使得把所有計程車的編號照著收費由小到大列出恰好是這個排列（當收費價格一樣，順序可以任意安排），若存在的話請輸出任意一組解，否則輸出「NIE」。
接著會有 q 筆修改，每筆修改會有 a_i, b_i，代表交換排列在 a_i 和 b_i 的數字，每次交換後皆須輸出先前問題的答案。**???**

- $1 \leq n \leq 5 \times 10^5$。
- $1 \leq q \leq 5 \times 10^5$。

Pattern – Taxis

題目 (Taxis，POI 2018)

現在有 n 台計程車編號 $1 \sim n$，對於第 i 台計程車，給定 s_i, c_i，代表該台計程車的收費方式為 $s_i + d \times c_i$，其中 d 為里程數。給定一個 $1 \sim n$ 的排列，請問是否存在一個里程數 x/y，使得把所有計程車的編號照著收費由小到大列出恰好是這個排列（當收費價格一樣，順序可以任意安排），若存在的話請輸出任意一組解，否則輸出「NIE」。

- $1 \leq n \leq 5 \times 10^5$。
- $1 \leq q \leq 5 \times 10^5$。

Pattern – Taxis

題目 (Taxis，POI 2018)

現在有 n 台計程車編號 $1 \sim n$，對於第 i 台計程車，給定 s_i, c_i，代表該台計程車的收費方式為 $s_i + d \times c_i$，其中 d 為里程數。給定一個 $1 \sim n$ 的排列，請問是否存在一個里程數 x/y，使得把所有計程車的編號照著收費由小到大列出**恰好是這個排列 ???**(當收費價格一樣，順序可以任意安排)，若存在的話請輸出任意一組解，否則輸出「NIE」。

- $1 \leq n \leq 5 \times 10^5$。
- $1 \leq q \leq 5 \times 10^5$。

Pattern – Taxis

什麼時候......

- 計程車照收費排序剛好是 $1, 2, \ldots, n$

Pattern – Taxis

什麼時候……

- ~~計程車照收費排序剛好是$1, 2, \ldots, n$~~

Pattern – Taxis

什麼時候……

- 🤮 ~~計程車照收費排序剛好是~~$\cancel{1, 2, \ldots, n}$
- 1 號車收費 $\leq$ 2 號，而且
- 2 號車收費 $\leq$ 3 號，而且
- ……
- $n-1$ 號車收費 $\leq n$ 號

Pattern – Taxis

什麼時候……

- ~~計程車照收費排序剛好是$1, 2, \ldots, n$~~
- 1 號車收費 $\leq$ 2 號，而且
- 2 號車收費 $\leq$ 3 號，而且
- ……
- $n-1$ 號車收費 $\leq n$ 號

$n = 2$ 我們總會做了吧？

Pattern – Taxis

只考慮 i 號車和 $i+1$ 號車，可以滿足「i 在 $i+1$ 前面」的距離 d 是一個 $d \leq \square$ 或 $d \geq \square$ 的限制

當每一組相鄰計程車的順序都滿足，所有車就會照順序排好

Pattern – Taxis

只考慮 i 號車和 $i+1$ 號車，可以滿足「i 在 $i+1$ 前面」的距離 d 是一個 $d \leq \square$ 或 $d \geq \square$ 的限制

當每一組相鄰計程車的順序都滿足，所有車就會照順序排好

維護一坨射線有沒有交集
可以用兩個 multiset 維護

Pattern – Taxis

帶修改？

「每筆修改會有 a_i, b_i，代表交換排列在 a_i 和 b_i 的數字」

Pattern – Taxis

帶修改？

~~「每筆修改會有a_i, b_i，代表交換排列在a_i 和b_i 的數字」~~

每次都做兩個單點修改

Pattern – Taxis

帶修改？

~~「每筆修改會有a_i, b_i，代表交換排列在a_i 和b_i 的數字」~~

每次都做兩個單點修改

時間複雜度：一次詢問 $O(1)$ 次 multiset 操作，總共 $O((n+q)\log n)$

Pattern – Grades

題目 (Grades，POI 2017)

有 n 位學生編號 $1 \sim n$ 以任意順序由左到右排成一列，現在你要派給這 n 位學生成績，成績必須是一個介於 $1 \sim n$ 之間的數字，且必須滿足以下條件：

- 若學生 u 的編號比學生 v 大，則學生 u 的成績不可以小於學生 v。
- 若學生 v 排在學生 u 右邊一位，則學生 v 的成績不可以小於學生 u，不然他會很傷心。

請問最多可以有多少不同的成績種類被派送出去？
接著會有 z 筆修改，每筆修改會有 p_i, q_i，代表交換排在位置 p_i 和 q_i 的學生編號，每次交換後皆須輸出先前問題的答案。

- $1 \leq n \leq 10^6$。
- $1 \leq z \leq 3 \times 10^5$。

Pattern – Grades

TL;DR

有 n 個學生從左到右排成一列，編號大的、排左邊的，成績要比較高。最多能派出幾種不同的成績？

Pattern – Grades

TL;DR

有 n 個學生從左到右排成一列，編號大的、排左邊的，成績要比較高。最多能派出幾種不同的成績？

z 次修改，每次交換隊伍裡兩個學生的位置

Pattern – Grades

TL;DR

有 n 個學生從左到右排成一列，編號大的、排左邊的，成績要比較高。最多能派出幾種不同的成績？

🙈 ~~z 次修改，每次交換隊伍裡兩個學生的位置~~

Pattern – Grades

$n = 2 \ldots\ldots$

- $[2, 1]$：兩種
- $[1, 2]$：一種

Pattern – Grades

有兩個人 $u < v$……

- v 排 u 左邊
- u 排 v 左邊

Pattern – Grades

有兩個人 $u < v$……

- v 排 u 左邊：v 的分數本來就該比較高
- u 排 v 左邊：$u, u+1, \ldots, v-1, v$ 的分數全都要一樣！

Pattern – Grades

看兩兩相鄰學生的編號關係，獲得（最多）$n-1$ 條限制「$u_i, u_i+1, \ldots, v_i-1, v_i$ 的分數要一樣」

Pattern – Grades

看兩兩相鄰學生的編號關係，獲得（最多）$n-1$ 條限制
「$u_i, u_i+1, \ldots, v_i-1, v_i$ 的分數要一樣」

要怎麼詢問「最多能派出幾種不同的成績」？

Pattern – Grades

看兩兩相鄰學生的編號關係，獲得（最多）$n-1$ 條限制「$u_i, u_i+1, \ldots, v_i-1, v_i$ 的分數要一樣」

每條限制都是「$u_i+1, \ldots, v_i-1, v_i$ 的分數都固定了，只有 u_i 的分數可以自由決定」
最後數數看有幾個人的分數可以自由決定

Pattern – Grades

看兩兩相鄰學生的編號關係，獲得（最多）$n-1$ 條限制
「$u_i, u_i+1, \ldots, v_i-1, v_i$ 的分數要一樣」

每條限制都是「$u_i+1, \ldots, v_i-1, v_i$ 的分數都固定了，只有 u_i 的分數可以自由決定」
最後數數看有幾個人的分數可以自由決定

區間加值（± 1），數數看全域有幾個 0

Pattern – Grades

帶修改（交換兩人位置）？

依然可以是兩次單點修改

Pattern – Grades

區間加值、單點修改、數全域有幾個 0

拿你最喜歡的資料結構砸掉
時間複雜度 $O((n+z)\log n)$

Pattern

我們是怎麼做完前面兩題的？

- 題目要維護的東西難以維護（整個排列的長相）
- 找到小小的特徵點，用小特徵湊出題目要的條件（排列中相鄰元素關係）
- 小特徵足夠單純可以維護（multiset、線段樹）

Pattern – Seats

題目 (Seats，IOI 2018)

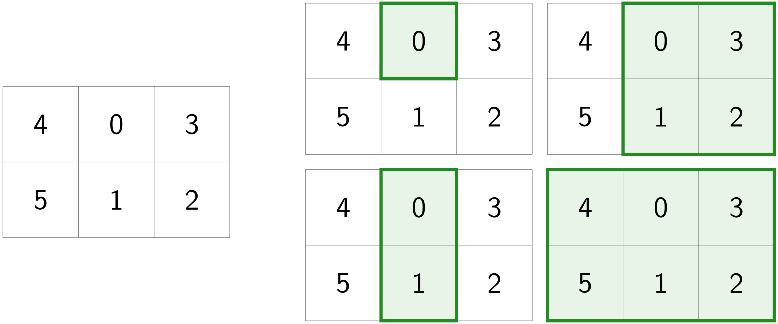
(完整敘述請見講義)

有 $H \times W$ 個位子排成一個矩形，還有 HW 位選手每人分別佔一個位置。有幾個**矩形**，使得矩形區域內坐的選手的編號恰好是 0 開始的連續數字？

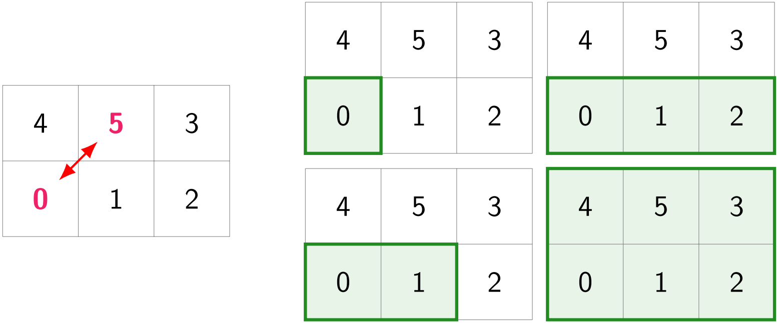
支援 Q 次修改，每次交換兩位選手的位置，每次交換完輸出以上問題的答案。

- $1 \leq H \times W \leq 10^6$
- $1 \leq Q \leq 5 \times 10^4$

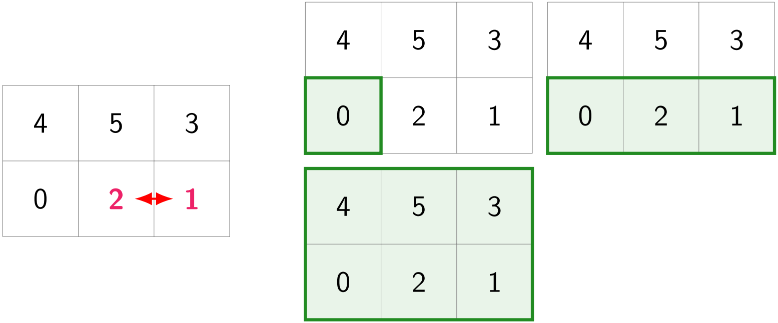
Pattern – Seats



Pattern – Seats



Pattern – Seats



Pattern – Seats

題目 (Seats，IOI 2018)

Subtasks

- (5) $HW \le 100$，$Q \le 5000$
- (6) $HW \le 10^4$，$Q \le 5000$
- (20) $H \le 1000$，$W \le 1000$，$Q \le 5000$
- (6) $Q \le 5000$，對於每次交換 $|a - b| \le 10^4$
- (33) $H = 1$
- (30) 無額外限制

拿零分還可以金牌的難題 !?

Pattern – Seats

選手一個一個坐進去，檢查他們是不是坐成矩形的樣子

躲不開的障礙：
要怎麼檢查一個矩形範圍是不是好的？
要怎麼檢查 $0, \ldots, rc - 1$ 的範圍是不是好的？

Pattern – Seats

題目 (Seats，IOI 2018)

Subtasks

- (5) $HW \leq 100$，$Q \leq 5000$
- (6) $HW \leq 10^4$，$Q \leq 5000$
- (20) $H \leq 1000$，$W \leq 1000$，$Q \leq 5000$
- (6) $Q \leq 5000$，對於每次交換 $|a - b| \leq 10^4$
- $\boxed{\textbf{(33)}\ H = 1}$
- (30) 無額外限制

二維太荒謬了，先想辦法搞定一維

Pattern – Seats

選手一個一個坐進去，檢查他們是不是坐成連續區間的樣子

躲不開的障礙：
要怎麼檢查一個區間是不是好的？
要怎麼檢查 $0, \ldots, rc-1$ 的範圍是不是好的？

Pattern – Seats

٩(ˊᗜˋ*)و

Pattern – Seats

把 $0, \ldots, rc-1$ 塗黑色，其他格子和界外留白。他們剛好在一個連續區間的**充要條件**是......

Pattern – Seats

把 $0, \ldots, rc-1$ 塗黑色，其他格子和界外留白。他們剛好在一個連續區間的**充要條件**是……

對於每一組相鄰的格子，**恰好有兩組是一黑一白**

Pattern – Seats

٩(ˊᗜˋ*)و

Pattern – Seats

把 $0, 1, \ldots, HW - 1$ 一個一個塗黑，每個時間點檢查是不是恰好兩組格子是一黑一白

Pattern – Seats

對於每一組相鄰的格子，他們在哪些時間是一黑一白？

Pattern – Seats

對於每一組相鄰的格子，他們在**某個連續的時間區間**是一黑一白

拿出你最喜歡的資料結構，維護每個時間點一黑一白的格子有幾組，數數看全域有幾個 2

Pattern – Seats

對於每一組相鄰的格子，他們在**某個連續的時間區間**是一黑一白

拿出你最喜歡的資料結構，維護每個時間點一黑一白的格子有幾組，**數數看全域有幾個 2 ???**

Pattern – Seats

對於每一組相鄰的格子，他們在**某個連續的時間區間**是一黑一白

只要有格子是黑的，一黑一白的格子就至少有兩組

拿出你最喜歡的資料結構，維護每個時間點一黑一白的格子有幾組，**檢查全域最小值是不是** 2、**數數看最小值有幾個**

Pattern – Seats

修改？還是可以兩次單點修改

時間複雜度：$O((W+Q)\log W)$

Pattern – Seats

題目 (Seats，IOI 2018)

Subtasks

- (5) $HW \leq 100$，$Q \leq 5000$
- (6) $HW \leq 10^4$，$Q \leq 5000$
- (20) $H \leq 1000$，$W \leq 1000$，$Q \leq 5000$
- (6) $Q \leq 5000$，對於每次交換 $|a-b| \leq 10^4$
- (33) $H = 1$
- **(30) 無額外限制**

Pattern – Seats

٩(´▽`*)و

Pattern – Seats

把 $0, \ldots, rc-1$ 塗黑色，其他格子和界外留白。他們剛好形成一個矩形的**充要條件**是......

Pattern – Seats

把 $0, \ldots, rc-1$ 塗黑色，其他格子和界外留白。他們剛好形成一個矩形的**充要條件**是......

對於每一塊 2×2 相鄰的格子，

- 恰好四塊是一黑三白

Pattern – Seats

٩(ˊᗜˋ*)و：甜甜圈

Pattern – Seats

把 $0, \ldots, rc-1$ 塗黑色，其他格子和界外留白。他們剛好形成一個矩形的**充要條件**是......

對於每一塊 2×2 相鄰的格子，

- 恰好 4 塊是一黑三白
- **沒有任何一塊是三黑一白**

Pattern – Seats

٩(´▽`*)و

Pattern – Seats

對於每一組 2×2 的格子，他們在哪些時間是一黑三白、或是三黑一白？

Pattern – Seats

對於每一組 2×2 的格子，他們在**某個連續的時間區間**是一黑三白、或是三黑一白

只要有格子是黑的，一黑三白的格子就至少有 4 組

拿出你最喜歡的資料結構，維護每個時間點一黑三白、三黑一白的格子有幾組，**檢查全域最小值是不是** 4**、數數看最小值有幾個**

Pattern – Seats

修改？還是可以兩次單點修改

時間複雜度：$O((HW + Q)\log HW)$
(多花點力氣 $O(HW + Q\log HW)$)

Pattern – Seats

修改？還是可以兩次單點修改

時間複雜度：$O(HW + \mathbf{16}Q \log HW)$
常數巨大！

Pattern – 好的連續子序列

題目 (好的連續子序列，台大演算法設計與分析（ADA）作業)

給定一個 $1, 2, \ldots, N$ 的排列，試求有多少子區間 $[l, r]$，滿足該子區間是一個連續正整數的排列？

- $1 \leq N \leq 5 \times 10^5$

在 Codeforces 526F 有可以傳的 Judge

Pattern – 好的連續子序列

跟一維的 Seats 比起來，少了修改，多了不是 1 開頭的區間也要數數看

Pattern – 好的連續子序列

跟一維的 Seats 比起來，少了修改，多了不是 1 開頭的區間也要數數看

- 用 Seats 作法找出 1 開頭的好區間有幾個
- 把 1 拿掉（讓他永遠是白色），找出 2 開頭的好區間有幾個
- ......
- 把 $1, 2, \ldots, N-1$ 拿掉（讓他們永遠是白色），找出 N 開頭的好區間有幾個

Pattern – 好的連續子序列

跟一維的 Seats 比起來，少了修改，多了不是 1 開頭的區間也要數數看

- 用 Seats 作法找出 1 開頭的好區間有幾個
- 把 1 拿掉（讓他永遠是白色），找出 2 開頭的好區間有幾個
- ……
- 把 $1, 2, \ldots, N-1$ 拿掉（讓他們永遠是白色），找出 N 開頭的好區間有幾個

題目沒叫你修改，但是你自己把「枚舉排列的開頭」當成 N 次修改

Pattern – 好的連續子序列

時間複雜度：$O(N \log N)$

Pattern – 好的連續子序列

時間複雜度：**至少** $O(4N \log N)$

常數巨大！我在 NEOJ 吃 TLE

Pattern – 好的連續子序列：番外

本題官方作法是分治，也有其他使用大資料結構但可以時限內通過的作法

你能想到幾種不同的作法？

Pattern – 好的連續子序列：番外

關鍵觀察：好區間 $\iff r - l = \max_{l \leq i \leq r}(a_i) - \min_{l \leq i \leq r}(a_i)$

Pattern – 好的連續子序列：番外

關鍵觀察：好區間 $\iff r - l = \max_{l \le i \le r}(a_i) - \min_{l \le i \le r}(a_i)$

分治作法（$O(N \log N)$）

- 序列切兩半，數跨兩邊的好的子序列
  - 分四種情況：最大值和最小值分別在左半邊還是右半邊
  - 時間複雜度 $O(N)$
- 兩邊遞迴分治

Pattern – 好的連續子序列：番外

關鍵觀察：好區間 $\iff r - l = \max_{l \le i \le r}(a_i) - \min_{l \le i \le r}(a_i)$

資料結構作法（$O(N \log N)$）

- 掃描線枚舉區間右界，用資結維護每個左界對應到的 $((\max - \min) - (r - l))$
  - $\max, \min$ 都可以單調隊列區間加值
  - 數數看有幾個 0
  - 這坨總是 ≥ 0，可以數最小值個數

Pattern

- 題目要維護的東西難以維護
- 找到小小的特徵點，用小特徵湊出題目要的條件
- 小特徵足夠單純可以維護

資結不是重點，重點是發現精妙的轉換和觀察

線段樹的暴力與懶人標記

線段樹的暴力與懶人標記

Change my mind：均攤分析就是玄學

線段樹的暴力與懶人標記 – 區間開根號

題目 (帶修改區間和，Zerojudge c652)

給你一段 N 個正整數的序列 $a_1, \cdots, a_N$，請你執行 Q 筆操作。

- 0 l r：代表詢問 $[l, r]$ 區間的和。
- 1 l r：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\lfloor \sqrt{a_i} \rfloor$。
- $1 \leq N, Q \leq 3 \times 10^5$。
- $1 \leq a_i \leq 10^{12}$。

線段樹的暴力與懶人標記 – 區間開根號

如果想在線段樹維護區間和
區間開根號的時候區間和會如何變化？

線段樹的暴力與懶人標記 – 區間開根號

如果想在線段樹維護區間和
區間開根號的時候區間和會如何變化？

我也不知道 😭

區間開根號沒有**可預測性**，不能打懶人標記

線段樹的暴力與懶人標記 – 區間開根號

觀察：一個數字被開 $\log\log C$ 次根號之後就不會再動了，永遠都會是 1

線段樹的暴力與懶人標記 – 區間開根號

觀察：一個數字被開 $\log\log C$ 次根號之後就不會再動了，永遠都會是 1

- 如果區間內有人不是 1，暴力往下修改
- 如果區間內所有人都是 1，什麼事都不需要做

一個數字只會被暴力改 $\log\log C$ 次，每次 $O(\log N)$ 時間
總時間複雜度 $O(Q\log N + N\log N\log\log C)$

線段樹的暴力與懶人標記 – 區間開根號・其二

題目 (帶修改區間和 Ex.，波路自編題)

給你一段 N 個正整數的序列 $a_1, \cdots, a_N$，請你執行 Q 筆操作。

- $1\ l\ r\ c$：**代表將 $[l, r]$ 區間的每個數字 a_i 加上 c。**
- $2\ l\ r$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\lfloor \sqrt{a_i} \rfloor$。
- $3\ l\ r$：代表詢問 $[l, r]$ 區間的和。

- $1 \le N, Q \le 10^5$。
- $0 \le a_i, c \le 10^9$。

線段樹的暴力與懶人標記 – 區間開根號・其二

開完根號再加值，一個數字只會被暴力改......$O(Q)$ 次（???)

剛剛的作法壞掉了

線段樹的暴力與懶人標記 – 區間開根號 · 其二

觀察：區間全距被開幾次根號之後就幾乎不動了

線段樹的暴力與懶人標記 – 區間開根號・其二

觀察：區間全距被開 $O(\log \log C)$ 次根號之後就會一直是 0 **或** 1

對全距是 1 的區間開根號可以打懶人標記嗎？
可以，多紀錄最小值和最小值個數就知道區間裡有哪些數字

線段樹修改時，只要全距還不是 1 就暴力往下修改

線段樹的暴力與懶人標記 – 區間開根號・其二

一開始，每個節點的全距都是 $O(C)$，暴力往下次數 $O(N \log\log C)$

每次區間加值讓 $O(\log N)$ 個節點的全距增加 $O(C)$，暴力往下次數增加 $O(\log N \log\log C)$

總時間複雜度 $O(Q \log N + N \log\log C + Q \log N \log\log C)$

線段樹的暴力與懶人標記

```
void modify(int node, int l, int r, int ql, int qr) {
    if(l >= ql && r <= qr) {
        give_tag(node); return;
    }
    push(node);
    int m = (l + r) / 2;
    if(ql <= m) modify(L(node), l, m, ql, qr);
    if(qr > m)  modify(R(node), m + 1, r, ql, qr);
    pull(node);
}
```

線段樹的暴力與懶人標記

```
void modify(int node, int l, int r, int ql, int qr) {
    if(l >= ql && r <= qr && 全距 <= 1) {
        give_tag(node); return;
    }
    push(node);
    int m = (l + r) / 2;
    if(ql <= m) modify(L(node), l, m, ql, qr);
    if(qr > m)  modify(R(node), m + 1, r, ql, qr);
    pull(node);
}
```

線段樹的暴力與懶人標記

```
void modify(int node, int l, int r, int ql, int qr) {
    if(tag_condition(node)) {
        give_tag(node); return;
    }
    push(node);
    int m = (l + r) / 2;
    if(ql <= m) modify(L(node), l, m, ql, qr);
    if(qr > m)  modify(R(node), m + 1, r, ql, qr);
    pull(node);
}
```

線段樹的暴力與懶人標記

也許......`tag_condition` 還可以是......？

線段樹的暴力與懶人標記 – Segment Tree Beats

題目 (Gorgeous Sequence，HDU 5306)

T 筆測資，每筆測資給你一段 N 個整數的序列 $a_1, \cdots, a_N$，請你執行 Q 筆操作。

- $0\ l\ r\ t$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\min(a_i, t)$。
- $1\ l\ r$：代表詢問 $[l, r]$ 區間的最大值。
- $2\ l\ r$：代表詢問 $[l, r]$ 區間的和。

- $1 \le T \le 100$。
- $1 \le \sum N, \sum Q \le 10^6$。
- $0 \le a_i, t < 2^{31}$。

區間取 min 對區間和同樣不能預測，不能直接打懶人標記

線段樹的暴力與懶人標記 – Segment Tree Beats

每個節點維護**區間嚴格次大值**和最大值個數

線段樹的暴力與懶人標記 – Segment Tree Beats

每個節點維護**區間嚴格次大值** ‼ 和最大值個數

- 如果 $t \leq$ 次大值，暴力往下修改
- 如果 $t >$ 次大值，等同於把所有最大值都改成 t，可以打懶人標記

時間複雜度：$O((N+Q)\log N)$

線段樹的暴力與懶人標記 – Segment Tree Beats

每個節點維護**區間嚴格次大值** !! 和最大值個數

- 如果 $t \leq$ 次大值，暴力往下修改
- 如果 $t >$ 次大值，等同於把所有最大值都改成 t，可以打懶人標記

時間複雜度：$O((N+Q)\log N)$???
憑什麼這麼快？

線段樹的暴力與懶人標記 – Segment Tree Beats

考慮每個節點的數字種類數
每次往下暴力修改，額外花 $O(1)$ 時間，區間內的數字一定會少至少一種

比一般線段樹多付出的時間
最多是每個節點暴力往下修改的總次數
也就是 $O(N \log N)$

線段樹的暴力與懶人標記 – Segment Tree Beats

考慮每個節點的數字種類數
每次往下暴力修改，額外花 $O(1)$ 時間，區間內的數字一定會少至少一種

比一般線段樹多付出的時間
最多是每個節點暴力往下修改的總次數
也就是 $O(N \log N)$

總時間複雜度 $O((N+Q) \log N)$

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

題目

給你一段 N 個整數的序列 $a_1, \cdots, a_N$，請你執行 Q 筆操作。

- $0\ l\ r\ t$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\min(a_i, t)$。
- $1\ l\ r\ c$：**代表將 $[l, r]$ 區間的每個數字加上 c。**
- $2\ l\ r$：代表詢問 $[l, r]$ 區間的最大值。
- $3\ l\ r$：代表詢問 $[l, r]$ 區間的和。
- $1 \leq N, Q \leq 3 \times 10^5$。
- $-10^6 \leq c, a_i, t \leq 10^6$。

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

嘗試跟前一題用一樣的作法

區間加值後，節點的數字種類數會變多……

嘗試跟前一題用一樣的作法

區間加值後，節點的數字種類數會變多……
O(區間長度)

沿用相同的證明想法，暴力修改的次數最多是……

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

嘗試跟前一題用一樣的作法

區間加值後，節點的數字種類數會變多......
O(區間長度)

沿用相同的證明想法，暴力修改的次數最多是......
$O(NQ)$？

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

嘗試跟前一題用一樣的作法

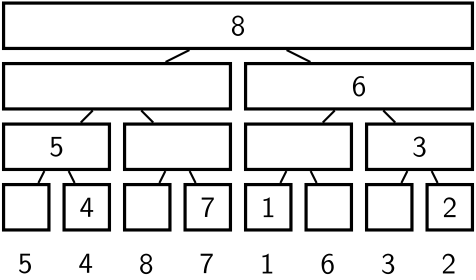
區間加值後，節點的數字種類數會變多......
O(區間長度)

沿用相同的證明想法，暴力修改的次數最多是......
$O(NQ)$？

換一種證明思路，可以證明總複雜度是 $O((N+Q)\log^2 N)$ 的

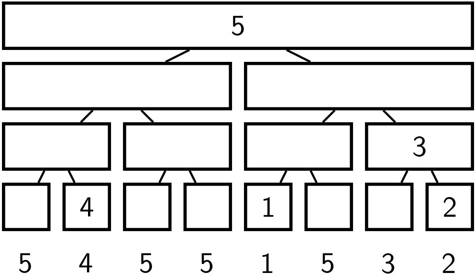
線段樹的暴力與懶人標記 – Segment Tree Beats・其二

序列 $[5, 4, 8, 7, 1, 6, 3, 2]$

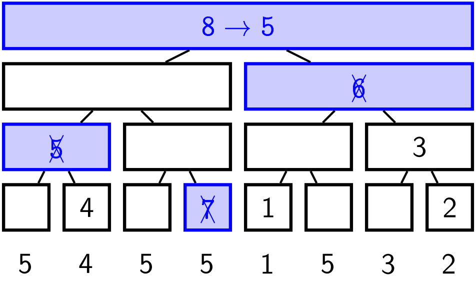


線段樹的暴力與懶人標記 – Segment Tree Beats・其二

$[1,6]$ 區間對 5 取 min

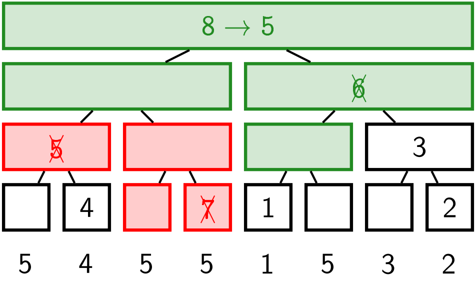


標記變不一樣的節點



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

修改過程中原本就會遞迴到的節點、和暴力往下修改的節點



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

「$t \leq$ 區間次小值時，往下暴力」
實際上等同往下 DFS 移除子樹內 $\geq t$ 的標記

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

「$t \leq$ 區間次小值時，往下暴力」
實際上等同往下 DFS 移除子樹內 $\geq t$ 的標記

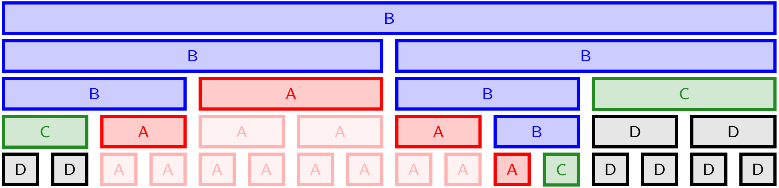
移除一個標記要花 $O(\text{樹高}) = O(\log N)$ 時間

一開始最多有 N 個標記
什麼時候標記會變多？

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

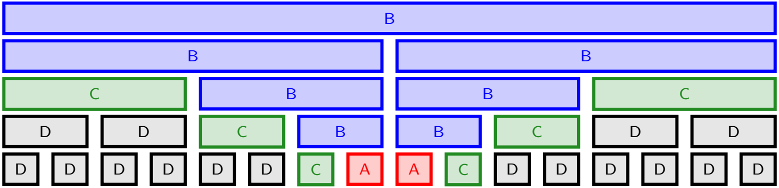
在線段樹上區間操作的時候，可以把節點分成四種

- A 被操作區間完全包含
- B 跟操作區間部份重疊
- C 跟操作區間不重疊，但是 B 的子節點
- D 跟操作區間不重疊的其他節點



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

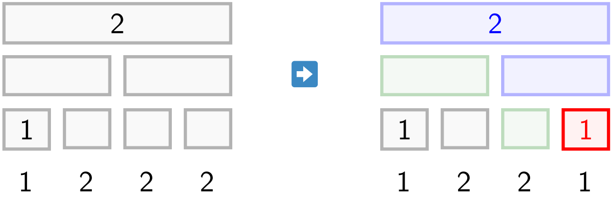
- B 類、C 類節點最多各 $O(\log N)$ 個
  A 類、D 類節點最多各 $O(N)$ 個
- A 類節點是 $O(\log N)$ 個子樹



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

標記變多例：A 類節點獲得標記（被操作區間完全包含）

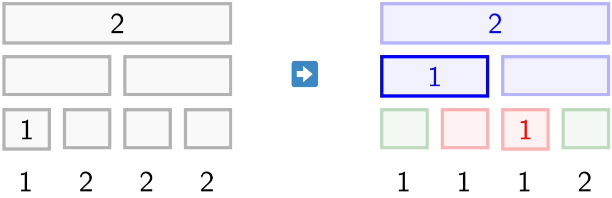
序列 $[1, 2, 2, 2]$
$[4, 4]$ 區間對 1 取 min



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

標記變多例：B 類節點獲得標記（跟操作區間部份重疊）

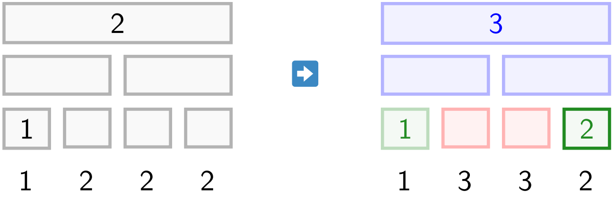
序列 $[1, 2, 2, 2]$
$[2, 3]$ 區間對 1 取 min



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

標記變多例：C 類節點獲得標記（跟操作區間不重疊，但是 B 的子節點）

序列 $[1, 2, 2, 2]$
$[2, 3]$ 區間加值 $+1$



線段樹的暴力與懶人標記 – Segment Tree Beats・其二

標記變多例：D 類節點獲得標記（跟操作區間不重疊的其他節點）

並不會，因為節點和父節點內的最大值都沒有變

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

場合 1：區間 chmin

- **A** 被操作區間完全包含
  - 減少若干個標記
  - 增加至多 $O(\log N)$ 個標記
- **B** 跟操作區間部份重疊
  - 至多 $O(\log N)$ 個節點
- **C** 跟操作區間不重疊，但是 B 的子節點
  - 至多 $O(\log N)$ 個節點
- **D** 跟操作區間不重疊的其他節點
  - 標記維持原狀

標記最多增加 $O(\log N)$ 個

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

場合 2：區間加值

A. 被操作區間完全包含
   - 標記維持原狀
B. 跟操作區間部份重疊
   - 至多 $O(\log N)$ 個節點
C. 跟操作區間不重疊，但是 B 的子節點
   - 至多 $O(\log N)$ 個節點
D. 跟操作區間不重疊的其他節點
   - 標記維持原狀

標記最多增加 $O(\log N)$ 個

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

- 「暴力往下」實際上是在 DFS 刪除標記
- 「暴力往下」刪除一個標記花 $O(\log N)$ 時間
- 總共只有 $O(N + Q \log N)$ 個標記可以刪

所以，吉如一線段樹和一般線段樹相比，額外花的時間頂多只有 $O((N+Q)\log^2 N)$

線段樹的暴力與懶人標記 – Segment Tree Beats・其二

吉如一本人給的證明和網路上流傳的證明都說可以 $O((N+Q)\log^2 N)$

實際上執行飛快，被懷疑其實只有一個 $\log$

線段樹的暴力與懶人標記 – Segment Tree Beats・其三

題目 (Range Chmin Chmax Add Range Sum，Library Checker)

給你一段 N 個整數的序列 $a_1, \cdots, a_N$，請你執行 Q 筆操作。

- $0\ l\ r\ t$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\min(a_i, t)$。
- $1\ l\ r\ t$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 $\max(a_i, t)$。
- $2\ l\ r\ c$：代表將 $[l, r]$ 區間的每個數字加上 c。
- $3\ l\ r$：代表詢問 $[l, r]$ 區間的和。

- $1 \leq N, Q \leq 3 \times 10^5$。
- $-10^6 \leq c, a_i, t \leq 10^6$。

線段樹的暴力與懶人標記 – Segment Tree Beats・其三

加上了區間取 max 操作

沿用同樣的作法同樣的證明，維護

- 區間最大、最小值
- 區間最大、最小值個數
- 區間**嚴格**次大、次小值

時間複雜度 $O((N+Q)\log^2 N)$

線段樹的暴力與懶人標記 – Bear and Bad Powers of 42

題目 (Bear and Bad Powers of 42，Codeforces 679E)

給你一段 N 個正整數的序列 $a_1, \cdots, a_N$，一個數字是好的若且唯若他不是 42 的冪次，請你執行 Q 筆操作。

- $1\ i$：輸出 a_i。
- $2\ l\ r\ x$：代表將 $[l, r]$ 區間的每個數字 a_i 改成 x，保證 x 是好的。
- $3\ l\ r\ c$：代表將 $[l, r]$ 區間的每個數字加上 c，並重複做**區間加值**直到 $[l, r]$ 區間的每個數字都是好的為止。

注意到每次操作後，所有數字都會是好的。

- $1 \leq N, Q \leq 10^5$。
- $2 \leq a_i, x \leq 10^9$。
- $1 \leq c \leq 10^9$。

「重複做區間加值直到 $[l, r]$ 區間的每個數字都是好的為止」？

例：序列 $[40, 41]$ 加值 $c = 1$

$[40, 41] \rightarrow [41, \mathbf{42}] \rightarrow [\mathbf{42}, 43] \rightarrow [43, 44]$

最終序列 $[43, 44]$

線段樹的暴力與懶人標記 – Bear and Bad Powers of 42

如果沒有區間改值，

- 一個數字頂多被加到 $NQ = 10^{14}$ 左右，而 10^{14} 以內的 42 冪次只有 $\log_{42} 10^{14}$ 不到十個
- 維護每個數字離下一個 42 冪次還有多遠

1. 區間減值
2. 區間詢問最小值
   - > 0？做完了，沒人壞掉，結束
   - $= 0$？有人剛好壞掉，單點改值，再區間減值一次
   - < 0？有人冪次變高，單點改值，再區間查最小值一次

時間複雜度 $O((N+Q)\log N \log_{42} NQ)$

如果加上區間改值，

- 不能暴力到底？暴力到什麼時候為止？

線段樹的暴力與懶人標記 – Bear and Bad Powers of 42

如果加上區間改值，

- 暴力到**區間內數字都一樣**為止
- 參考吉如一線段樹的證明，需要暴力很多次的節點不會增加很多
- 時間複雜度 $O((N+Q)\log N \log_{42} NQ)$

線段樹的暴力與懶人標記 – Bear and Bad Powers of 42

如果加上區間改值，

- 暴力到**區間內數字都一樣**為止
- 參考吉如一線段樹的證明，需要暴力很多次的節點不會增加很多
- 時間複雜度 $O((N+Q)\log N\log_{42}NQ)$???

線段樹的暴力與懶人標記 – Bear and Bad Powers of 42

題外話：官解

- 被區間改值的那段數字視為「一坨」
- 區間操作的時候可能把一坨切成兩坨
- 一整坨可以一起加值
- 看起來像 treap，不過可以用線段樹實做
- 時間複雜度 $O((N+Q)\log N \log_{42} NQ)$

線段樹的暴力與懶人標記

Change my mind：均攤分析就是玄學

線段樹的暴力與懶人標記 – 總結

這不是一堂資料結構課，這是一堂**均攤分析**課
資料結構不是重點，重點是均攤的思路、直覺、證明手法

也許你此生沒機會真的砸吉如一線段樹，但均攤分析值得你學習

結語

結語

砸資料結構的機會可能比你想像的更少，但在某個關鍵時刻，資料結構也許可以幫你在最後一步把題目車過去

結語

祝大家永遠都不需要用到進階資料結構