

Toj Theme Contest 001 資料結構入門

Toj Theme Contest 001 資料結構入門 試題本

競賽規則

1. 競賽時間：2025/03/30 14:00 ~ 16:00，共 2 小時。
2. 本次競賽試題共 5 題，每題皆有子任務。
3. 為了愛護地球，本次競賽題本僅提供電子檔，不提供紙本。
4. 每題的分數為該題所有子任務得分數加總；單筆子任務得分數為各筆繳交在該筆得到的最大分數。
5. 全部題目的輸入皆為標準輸入。
6. 全部題目的輸出皆為標準輸出。
7. 所有輸入輸出請嚴格遵守題目要求，多或少的換行及空格皆有可能造成裁判系統判斷為答案錯誤。
8. 每題每次上傳間隔為 30 秒，裁判得視情況調整。
9. 所有試題相關問題請於競賽系統中提問，題目相關公告也會公告於競賽系統，請密切注意。
10. 競賽中請勿交談。請勿做出任何會干擾競賽的行為。
11. 如需使用 C++ 的 `std::cin` 或 `std::cout` 可將以下程式碼插入 `main` function 以及將 `endl` 取代為 `'\n'` 來優化輸入輸出速度。唯須注意不可與 `cstdio` 混用。

```
std::ios::sync_with_stdio(false);
std::cin.tie(nullptr);
```

A. 有很多牌就能榮和

Problem ID: 2025TTC_Many

Time Limit: 1.0s

Memory Limit: 512MiB

— 本題輸入方式為非典型，請注意!!! —



Figure 1: 四暗刻

Blame，一位年輕的麻將新手，剛踏入這個充滿計算與推理的世界。他眼中的麻將桌，每一張牌不僅僅是一個數字，而是智慧、策略與運氣緊密相連的符號。這段旅程的起點，是因為 Blame 被台南一中資訊社的夥伴們吸引，看著他們每天中午聚在一起快樂對局，雖然偶爾有人眉頭微蹙，但整體氛圍依舊歡愉。而那一瞬間，當牌局節奏突然加快、有人胡出“四暗刻”，螢幕上絢麗的特效與驚人的點數展示，令 Blame 瞬間神魂顛倒。

這一幕深深地觸動了 Blame 的心，他決定也要加入這麻將的世界，並以最快的速度晉升為高手。雖然起初他對規則不甚了解，常因操作失誤而「放槍」，但很快他發現，一局麻將的勝負，不僅僅靠運氣，更需要在極短的反應時間內進行快速判斷與計算。於是，Blame 不再僅憑運氣，而是將目光放在一項關鍵技術上：只要能夠迅速掌握牌堆中出現次數最多的牌，就能為自己的策略提供強大支援，進而掌握勝利的先機。

然而，現實並不如想像中那般簡單。每局對戰中，牌堆內的牌數千變萬化，儘管每張牌代表的數字皆不相同，但數字卻可能極為龐大，甚至有時候出現的張數高達難以置信的數量。面對如此快速且變化莫測的局勢，Blame 的腦海忙亂運轉，卻難以在短暫的時間內逐一計算各牌的出現次數。於是，他決定求助於智慧的力量，設計出一套能夠迅速運算、動態更新的系統。

這個系統能夠處理多達 (Q) 次操作，每一次操作都會讓牌堆狀況發生改變：

1. 當一張牌（其牌面數字為 (A)）被放入牌堆時，程式會自動記錄並更新這張牌的數量。

2. 當一張牌（牌面為 (A)）從牌堆中被取出時，系統同樣會調整該牌的數量。
3. 當 Blame 查詢牌堆中出現最多次數的牌時，程式會立即回傳該牌的數字以及它在牌堆中的數量。如果多種牌的數量相同，則會回傳數字最小的一組。

這個系統不僅讓 Blame 能夠隨時掌握牌堆的動態，更在極短的反應時間內，提供他所需的關鍵資訊。牌堆中哪一種牌出現得最多，系統會在瞬間告知，並告訴他具體數量，從而協助他迅速作出策略調整。

這套簡單而高效的系統，迅速成為 Blame 在《雀魂麻將》中的制勝利器。曾經那個迷失方向的新手，逐漸變得從容、冷靜且充滿智慧。在每一局快節奏的對戰中，他不再只是依賴運氣，而是用精準的數據判斷，將每一步走得更為堅實。隨著時間推進，Blame 也漸漸領悟到麻將的真髓——它不僅是一場運氣與技巧的較量，更是一場關於邏輯、數據與快速反應的智慧之戰。

從此，Blame 的麻將人生因這個程式而煥然一新，他不再是那個在牌桌上徬徨的小白，而是以科技與智慧佔據了優勢。每一次操作、每一個決策，都在證明著他從失誤中學習、從挑戰中成長。這段充滿挑戰與機遇的旅程，不僅見證了他對麻將藝術的追求，更象徵著一位年輕人對於智慧與技術的無限熱情與堅持。

— 作答方式 —

本題採用函數呼叫方式。

請實作以下三個函式：

```
void add(int x) {  
    // 給你  $x$  要把  $x$  加入進系統當中  
}  
  
void remove(int x) {  
    // 給你  $x$  要把  $x$  從系統中刪除  
    // 假如  $x$  不存在系統中，請直接回傳  
}  
  
pair<int, int> query() {  
    // 詢問當前系統中的眾數值以及數量  
    // 假如有相同的請輸出值最小的那個  
    // 假如系統中沒東西，回傳  $\{-87, -87\}$   
}
```

— 輸入限制 —

- $1 \leq x \leq 10^9$
- $1 \leq \text{函式呼叫次數} \leq 10^5$

— 子任務 —

編號	分數	額外限制
0	100	無額外限制

B. 我是在進步還是退步？

Problem ID: 2025TTC_Progress

Time Limit: 1.0s

Memory Limit: 512MiB

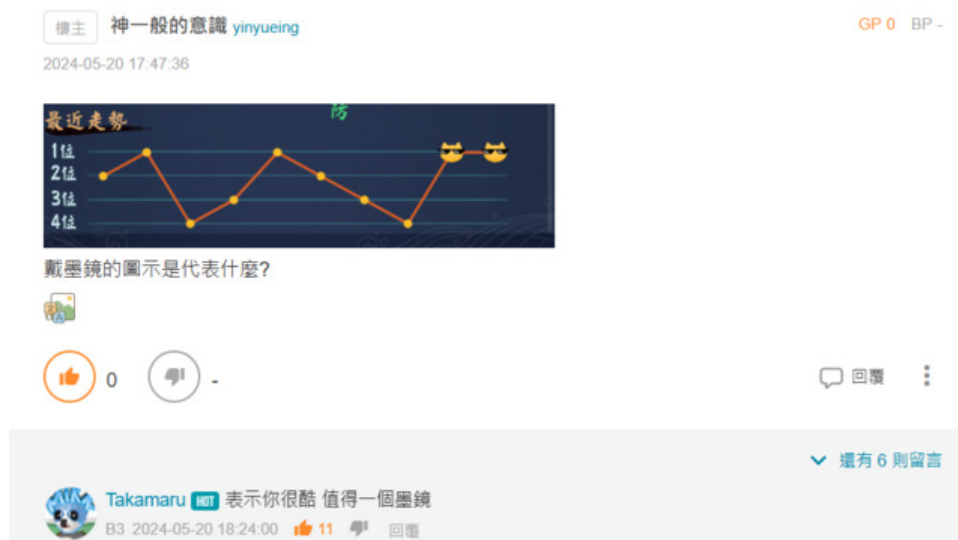


Figure 1: 從巴哈上截下來的某一帖

隨著時間流逝，Blame 的心神也逐漸被雀魂這款遊戲所吞噬。

一場又一場對局，他投入全神貫注，試圖用冷靜與計算掌控局勢，然而結局卻總是無情。

每當牌局結束，他都只能盯著那殘酷的結算畫面，看著自己被狠狠擊飛，失去分數，失去排名，甚至失去信心。

這樣的畫面一次又一次地上演，如同無形的錘子，不斷敲擊著他的心靈。

原本堅定的心態，在這些重擊下逐漸出現裂縫。

「是不是自己真的不適合？」、「是不是永遠都追不上別人？」

這些負面聲音在腦中盤旋不去，讓他難以喘息。

終於，在一場又一場的挫敗之後，Blame 徹底破防了。

他回到遊戲大廳，靜靜地盯著戰績頁面。

那上下起伏的折線圖，像是一條嘲諷的曲線，刻劃出他的不穩定與無力。

看著那條曲線，他感受到一股難以言喻的挫敗感，深深地壓在心頭。

明明努力過，明明想贏，明明曾經相信只要繼續學習和練習就能變強.....

但此刻，他卻只能無聲地坐在螢幕前，任由心中的自我質疑與失落感漸漸蔓延，淹沒了他最初的熱情與自信。

然而，當他盯著那起伏不定的戰績曲線時，忽然有個想法在腦海中閃過。

他開始細細翻閱自己的每場對局紀錄，不再只是盲目感受挫敗，而是嘗試從數據裡尋找規律。

他開始計算，記錄每一局的順位、得失分、對手強度，甚至連自己出現失誤的回合也標註了下來。

隨著一場場地分析，他漸漸發現了一個現象——自己的表現其實並非毫無進步，而是像一條排隊的隊伍，有起有落。

有時雖然連續低迷，但也偶爾會有高光時刻。

這時，他想到了自己曾經做過的一道經典題目：滑動窗口最大值。

那是一個看似簡單，卻蘊含深意的問題。

題目中，窗口可以透過三種操作改變：

第一種操作，是將一個元素丟進窗口的右端，意味著把新的一場戰績加入考量，就像是下一場對局的結果，無論好壞，都要納入這段旅程中；這時，窗口的右界往右延伸，序列變長，也多了一次機會改變走勢。

第二種操作，是將位於窗口左端的元素拔除，代表著忘記那些過去的低谷或高峰，專注於當下與未來；這就像將一段時間前的表現從評估中移除，窗口的左界向右移動，讓眼界更聚焦於最新的一段過程；不過，題目規定，如果窗口已經為空，就不操作，提醒著他：無論如何，必須手中至少有一些東西作為判斷依據，不能一無所有。

第三種操作，則是查詢目前窗口中的最大值。

這是一種充滿智慧的行為——不只看眼前的失敗或勝利，而是從一段區間裡找出那個「最高光」的時刻，提醒自己「我可以的」，「我曾經做到過」，「這並不是空想」。

最大值就像信心的燈塔，指引著方向，提醒著自己，即便目前是低潮，但只要這個滑動窗口裡還存在過高峰，那麼未來就還有機會再度出現高峰。

Blame 突然明白，雀魂也正是這樣。

人生本身，也是一道滑動窗口最大值問題。

他不需要被一時的低潮壓垮，也不需要執著於過去的失誤；

而是應該讓自己不斷向前延伸，記錄新的成就，捨棄過去的包袱，並在每個當下去尋找和期待那屬於自己窗口中的「最大值」。

當他意識到這一點，心態竟莫名地平靜了下來。

他重新打開對局大廳，輕輕點下「開始對戰」。

不再只是盯著勝負的結果，而是帶著計算者的理性與期待，把每一場對局當成往窗口中新增的一個元素，告訴自己：

「沒關係，這只是滑動窗口的一部分。」

「下一次，那個最大值，一定會屬於我。」

— 輸入 —

第一行輸入一數 Q

接下來 Q 行

每行開頭會有一數 X 代表操作種類

假如為操作 1，後面會接著一個數字 A

— 輸出 —

根據操作 3 輸出答案，當窗口中並無任何元素時，請直接輸出“-1”即可。

— 輸入限制 —

- $1 \leq Q \leq 10^6$
- $X \in \{1, 2, 3\}$
- $1 \leq A \leq 2^{31} - 1$
- 操作 3 必定會出現一次以上

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	17	$1 \leq Q \leq 10^3$
3	43	$1 \leq Q \leq 10^5$
4	40	無額外限制

— 範例輸入 —

```
7
1 87
1 4
3
2
3
2
3
```

— 範例輸出 —

```
87
4
-1
```


C. 怎麼有人一直在天上飛?

Problem ID: 2025TTC_Flying

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 從巴哈上偷來的一張心酸圖

自從 Blame 逐漸適應了在雀魂麻將中摸滾打爬的生活後，他早已深深融入這片充滿競爭和挑戰的麻將江湖。每當他坐在麻將桌前，聽著牌面發出清脆的聲音，心中總有一股莫名的熱血澎湃。這場遊戲不僅僅是一場策略的比拼，更像是一場關乎智慧與運氣的冒險，每一局都充滿了驚心動魄的變數。在這條旅程中，Blame 不僅磨練了自己的牌技，還逐漸明白了雀魂中每個玩家背後的不同心態與人生哲學。

然而，當雀魂推出了一款全新的多人混戰模式時，Blame 的生活再次發生了變化。這款模式將 N 個玩家圍成一圈，從第一位玩家開始，順時針進行遊戲。每一位玩家都有各自獨特的個性，他們的行為不僅影響著遊戲的進程，還暗示著他們對勝利的渴望和對挑戰的態度。這些玩家有著 Blame 所熟悉的三種典型性格，他們之間的博弈充滿了智慧與謀略，而每一場對局都像是一場人生的縮影。

這三種性格的玩家，無一不在每一局中展現出不同的特徵。每當輪到他們時，他們的選擇、出牌方式，甚至是眼神和動作，都透露出他們對遊戲的理解和對勝利的渴望。在這場充滿陰謀與策略的混戰中，Blame 發現了玩家們的真實面貌。

雷霆戰魂

首先，Blame 看到了第一類玩家，他們正是所謂的**雷霆戰魂**。每一場遊戲對這些玩家來說，都是一場殊死戰。他們的眼中只有目標，心中只有勝利的渴望。他們絲毫不會顧及周圍人的感受，甚至不會去想其他玩家的策略和未來的計劃。每當輪到他們時，他們就會毫不猶豫地出牌，準確

無誤地胡出役滿，將下家（下一位玩家）一舉淘汰。這種無情的淘汰方式讓整個遊戲充滿了暴力的氣息。他們的每一次出牌都如同一記致命的雷霆，準確地將敵人擊飛，徹底摧毀對方的希望。這些玩家從不拖延，從不猶豫，總是以最迅速的速度和最果斷的手段，將對手踢出局外，留下自己繼續向前。

雷霆戰魂的每一場遊戲，彷彿都是一次蓄勢待發的攻擊。他們不會為了體面或顧及其他人，而是將每一次機會都視為戰爭的起點，當下家的一聲敗落，他們便會冷靜地看著自己站上頂端，無懈可擊。他們的目標只有一個——在這場混亂的戰場中，站到最後。

暗影謀士

而與此同時，Blame 發現了第二類玩家——**暗影謀士**。他們的遊戲方式比雷霆戰魂更為精妙，也更加陰險。這些玩家的每一個動作都充滿了深思熟慮，他們會觀察每一位對手，琢磨每一個細節，最終在適當的時候出手。當輪到他們時，他們並不會直接攻擊，而是用一種十分隱蔽的方式，悄無聲息地斷除對手的希望。對他們而言，勝利並非來自一場猛攻，而是來自精心策劃的計算與耐心。他們的每一步棋，仿佛是一個精心策劃的陰謀，充滿了狡猾的智慧和高超的計謀。這些玩家總是能在最不經意的時候，完成最具殺傷力的行動，將對手拖入無法自拔的困境。

暗影謀士的行為不僅僅是為了自己能活下來，而是為了將整個遊戲都操控在自己的手中。他們清楚地知道，勝利並不總是來自一場華麗的勝利，而是來自於將每一位對手的計劃拆解，並在最適當的時候，將他們一一擊敗。他們不會顯示出任何的不安與焦慮，反而總是以冷靜的態度，暗中撬動局勢，讓自己的勝利如影隨形。

隱忍求生

而第三類玩家，則是那群**隱忍求生**型人物。他們對遊戲的態度非常不同於前兩者。當輪到他們時，他們並不會急於出牌，也不會計劃一場華麗的反擊。相反，他們會選擇打出一張又一張的安全牌，確保自己不會因為冒險而陷入困境。雖然他們的方式看似保守，但也正是這種穩妥的策略讓他們得以在其他玩家相互淘汰的過程中，安穩地存活到最後。他們並不追求快速的勝利，也不渴望一場華麗的翻盤，他們的目標只有一個——在混亂的遊戲中穩穩地活下來，活到最後的那一刻。這些玩家的策略也許看似消極，但他們深知，只有在其他人相繼倒下的時候，才是他們站上巔峰的時候。

隱忍求生的玩家，或許永遠無法成為遊戲中的耀眼明星，但他們總能在最危險的時刻，保住自己的生命。他們不需要驚天動地的表現，也不會追逐炫目的結果。對他們來說，麻將的意義不在於一場華麗的勝利，而是在於穩穩地過完每一局，最終成為最後的幸存者。

這三類玩家的性格與行為，構成了雀魂中獨特的競技氛圍。他們或是如雷霆般的強勢，或是如暗影般的深謀，或是如樹林中的獵物般隱匿。而 Blame 也在這樣的競技場中，發現了自己的人生哲學：無論是哪一種性格，每一種方式都有其值得學習的地方，而最終能夠存活下來的，往往是那些最能適應變化、最懂得因應局勢的人。

在這場混戰的遊戲中，Blame 目睹了每位玩家的智慧和行為，逐漸明白了一個道理：無論多麼強勢的玩家，無論多麼隱秘的計謀，最終贏得這場遊戲的，往往是那些能夠在關鍵時刻適應變化、靈活應對的人。

現在，假如給定了 N 個玩家（編號 $1 \sim N$ ， $1 \rightarrow 2, 2 \rightarrow 3, \dots, N \rightarrow 1$ ），並且每位玩家的性格屬性分別為 A、B、C，Blame 心中有一個疑問：這場混戰，最終的勝者會是誰呢？

在這場遊戲中，每個玩家有三種可能的行為：

- **雷霆戰魂**（A 型玩家）：每當輪到他們時，他們會毫不猶豫地出牌，準確無誤地將下家擊飛，淘汰出局。
- **暗影謀士**（B 型玩家）：他們會選擇在合適的時候發動致命一擊，將上家淘汰出局。
- **隱忍求生**（C 型玩家）：這些玩家則總是選擇安全牌，不冒險地過關，努力在每一局中生還。

每輪的遊戲都依循著固定的規則進行：當某一名玩家被擊飛後，他的上家會成為下家，並且他的下家會繼續輪流進行。每一次玩家的淘汰，都會改變局勢，可能引發一連串的變化，直到只剩下最後的幸存者。

這時，Blame 想問：最終的贏家會是誰呢？他想知道，在這場動盪不安的局中，究竟是哪一位玩家能夠憑藉自己的獨特策略，最終笑到最後。

— 輸入 —

第一行輸入一數 N

第二行輸入一長度為 N 的字串 S

代表著編號 $1 \sim N$ 的玩家性格屬性

— 輸出 —

輸出最後活著的人的編號。

— 輸入限制 —

- $1 \leq N \leq 10^6$
- $S_i \in \{A, B, C\}, 1 \leq i \leq N$
- 保證最後只會留下來一人。

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	3	$S_i \in \{A\}, 1 \leq i \leq N$
3	4	$S_i \in \{B\}, 1 \leq i \leq N$
4	25	$1 \leq N \leq 10^4$
5	68	無額外限制

— 範例輸入 1 —

5
ABCAB

— 範例輸出 1 —

4

— 範例輸入 2 —

3
BAA

— 範例輸出 2 —

2

— 範例輸入 3 —

7
ACCCCC

— 範例輸出 3 —

1

D. 不要吵架了啦!!!

Problem ID: 2025TTC_Fight

Time Limit: 1.0s

Memory Limit: 512MiB



Figure 1: 嘿嘿嘿

Blame 最近沉迷雀魂，整天泡在遊戲裡，不僅課金抽角，還花上無數時間替每位女角挑選服裝。
「這件和式小袖配這個髮飾，剛剛好。」
「這雙鞋子，會不會太高調？還是換個淡雅點的吧。」

每天晚上，他都會靜靜坐在電腦前，反覆在衣櫥與角色頁面之間切換，
從髮飾到指甲油、從圍巾到耳環，一絲不苟，彷彿在替真正的朋友打點舞會的行頭。

他的收藏裡有 N 名女角，每一位都有獨特的個性，或溫柔、或傲嬌、或冷淡、或活潑，
衣櫥裡則堆滿了 M 種風格各異的服裝，宛如一間私人精品店。
那裡有刺繡和服、簡約西裝、古典洋裝、甚至還有童話風的禮服，每一件都閃爍著 Blame 細心
與心血的光芒。

最近，雀魂推出了一場盛大的「友人約戰」活動。
Blame 激動地將這消息分享至每個群組，心想這一次，終於能讓自己親手打扮的女角們同台展
現風采。

然而，很快，他便發現事情遠比想像中複雜。
在這場活動中，遊戲系統會隨機丟出「摩擦事件」——每當事件觸發，代表兩位女角之間產生了
微妙的敵意。
她們會暗中較勁、不滿，甚至冷戰。
而在最終服裝分配階段，如果有任意一對摩擦中的女角被分配到相同風格的服裝，
那場比賽便會瞬間崩潰，舞台在一片爭吵聲中落幕，連 Blame 也只能無力地望著失敗訊息。

起初，摩擦事件只有零星幾條。

Blame 還能一邊喝著熱茶，一邊耐心地調整。

他就像解謎高手，手指飛快點選、拖曳，

仔細避開每一條紅線，

將服裝與女角巧妙搭配，讓大家和諧共處。

但當事件逐漸堆積、關係網越來越複雜時，問題開始爆炸。

剛調好這組，系統又彈出一條「小羽與小梅摩擦加劇」；

才換好另一組，下一條通知又說「夏音與美月發生矛盾」。

有幾次，Blame 幾乎崩潰。

他反覆嘗試上百種排列組合，

在腦海中畫下無數條線、無數張圖，

試圖讓所有有敵意的女孩都穿著不同風格的衣裳。

然而，終究還是失敗。

螢幕閃爍著系統提示：「目前狀態無法平衡。建議使用撤回事件功能。」

那個按鈕在深夜的螢光中不斷閃爍，就像催命符一般。

Blame 頓了頓，深吸一口氣，手指顫抖著按了下去。

精神力數值從 C 減去一點點，

那是他珍惜的資源，每次消耗都像是將自己的一點靈魂送進深淵。

撤回，調整，嘗試，再失敗。

撤回，再調整，再嘗試。

夜裡，他的雙眼充血，眼角泛紅，

桌上的水杯早已冷掉，

手心泛著細汗，卻依然不肯放棄。

「她們值得最好的舞台。」

「就算要耗盡精神力，也不能讓她們出糗。」

終於，在一次又一次近乎偏執的微調中，

他成功了。

所有女角在最終舞台上，各自穿著獨特卻完美契合的服裝，

沒有爭吵，沒有摩擦，

只有閃耀的聚光燈、輕盈的步伐、與完美和諧的微笑。

活動結束時，系統提示：

「恭喜完成所有分配。」

總共消耗精神力： X

謝謝您的努力。」

Blame 呆坐在螢幕前，疲憊到無法思考。

但當他看著螢幕上每位女角的照片時，

他忍不住笑了。

那一刻，他終於明白——那不只是精神力的消耗，而是他對美好與平衡的堅持。

就算只是遊戲，就算只是虛擬角色，

他仍願意傾盡全力，

只為讓這舞台，完美無瑕。

— 輸入 —

第一行輸入三數 N 、 M 、 C

第二行輸入一數 Q 代表總共發生的衝突次數

接下來輸入 Q 行，每行都有兩數 A 、 B ，代表角色 A 與角色 B 之間發生了摩擦事件。

— 輸出 —

輸出一數 X

— 輸入限制 —

- $2 \leq N \leq 10^5$
- $2 \leq M \leq 10^3$
- $1 \leq C \leq 87$
- $1 \leq Q \leq 3 \cdot 10^5$
- $1 \leq A, B \leq N$
- $A \neq B$
- M 是偶數

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	10	$1 \leq Q \leq 10^2, 2 \leq M \leq 8$
3	11	$1 \leq Q \leq 10^3$
4	20	$2 \leq N \leq 5 \cdot 10^2, 1 \leq Q \leq 10^4$
5	59	無額外限制

— 範例輸入 —

4 2 87

4

1 2

2 3

3 1

3 4

— 範例輸出 —

87