高一排名賽題解

出題者 : Samuel、Same0620、becaido

pA 水題

出題者 : Same0620

實體賽 First AC : RickRolling

線上賽 First AC : temmieowo

子題一：無其他限制

由於已保證每個水龍頭開啟的時間不遞減，因此只要在每次加入新水龍頭時維護此時每單位時間的流量，即可判斷在下一個水龍頭打開之前，是否會達到所需的容量。若否，則計算在下一個水龍頭開啟前會多出多少的水量；反之，則計算多少時間後會達到所需的容量。

pB 電學大師

出題者：Samuel

實體賽 First AC：iantsai0616

線上賽 First AC：temmieowo

子題三：無其他限制

應該可以直接講整題的，這題就是沒有包裝的DSU實作

如果不熟悉的可以去了解一下併查集(DSU)這個資料結構

他可以處理圖的連通性問題，且兩集合合併以及查詢所屬集合的複雜度皆為O(log n)

因此整體複雜度為O(q log n)

子題一：n<=1000

這筆預設的做法是開n*n的表紀錄兩兩是否連通

每次合併時，如果兩點並未連通，就將兩點原先連通的點兩兩連起來

複雜度O(n^2+q)

子題二：保證指令0後不會再出現指令1

允許將圖進行一次dfs處理連通性

簡單來說就是跑dfs判斷誰是同一組

複雜度O(n+q)

pC 臨時抱佛腳

出題者：Samuel

實體賽 First AC：lifeadventurer

線上賽 First AC：temmieowo

子題一：m=1

因為只有一個科目，變成要判斷所有難度嚴格遞增或嚴格遞減的序列最長有多長

其實可以直接用set找不同的數字個數

複雜度O(n log n)

子題二：n<=1000

要做到這個子題應該就必須有察覺這是LIS問題了

如果你有做出來，但沒有察覺的話

就代表你做了LIS但你沒發現，可以去熟悉最長遞增子序列

這題與LIS的關係是，當我們利用pair或其他形式將每本書對應的科目喜好與難度綁起來

並且進行排序後(先依科目喜好程度，再依難度遞增或遞減)

此時書本的科目喜好順序一定能滿足題目要求

再來再尋找難度嚴格遞增或嚴格遞減的最長子序列就好

本子題複雜度允許O(n^2)，但並不是LIS的最好作法

子題三：無其他限制

呈上子題，其實就是做LIS的O(n log n)作法

pD 油漆與底線

出題者 : Same0620

實體賽 First AC : iantsai0616

線上賽 First AC : Aestivate

子題一：n, len <= 1000

可以發現其實這題就是要區間改值以及單點查詢，而一次操作最多更改 len 個值，因此這筆子題可以直接用陣列暴力修改

複雜度：O(n * len)

子題二：無其他限制

用一個 set 存每個區間的左界以及顏色的數值，對於每筆操作，若左界和右界會覆蓋到另一個顏色的區間，將被覆蓋的區間分裂成會被覆蓋和不會被覆蓋的部分，並將所有被覆蓋的區間 erase 後，insert 該次操作新增的區間

複雜度：O(n log(len))

pE 下棋

出題者 : Same0620

實體賽 First AC :

線上賽 First AC : Penguin07

子題一：a, b, c, d <= 1000

不難發現，對於雙方而言，最佳的策略都是選花費最少的棋子下。由於對於給定範圍，(x, y) 最多只有 1000000 對，因此暴力枚舉後，透過輾轉相除法求最大公因數並記錄即可。令 cnt[i] 代表最大公因數為 i 的點的個數，讓 i 從小跑到大即可求出答案。

複雜度：O(n*m*log(max(a, b, c, d))) (n, m為矩形的長和寬)

子題二：無其他限制

令 dp[i] 代表最大公因數為 i 的個數，不難發現我們可以透過直接相除來判斷區間內有幾個 i 的倍數，由於 x 座標會界於 a ~ c 之間而 y 座標會界於 b ~ d 之間，因此 x 座標是 i 的倍數的個數一共有 c/i - (a-1)/i 個，同理可推知 y 座標中 i 的倍數的個數。由此觀察，我們可以算出一共有多少個點的 x, y 座標的含有 i 這個公因數並將這個數字存到 dp[i] 中。但這樣 dp[i] 中會包含到最大公因數為 2i, 3i, … 的點對，因此需要排容，從 dp[i] 中扣除 dp[2i], dp[3i], …，最後即為所求。剩下的部分和子題一相同，由 dp 的表格求出答案即可。

複雜度：O(n log(n)) (n 為 max(a, b, c, d))

pF 跑腿大師

出題者 : Samuel

實體賽 First AC : iantsai0616

線上賽 First AC : Aestivate

前情提要

在解任何一個子任務前，應該要先能把這題要問的重點給整理出來

首先問題可以簡化成在樹上，找到 l,l+1,...,r 的路徑中，路徑總長與 i 到 i+1的最大距離

再來我們可以再將其拆解為兩個部分

第一是如何求出兩點間的路徑長，第二是如何快速處理區間和與區間最大值

子題二：鍊

先講這個子題，他應該是最簡單的，都說是鏈了，點的編號甚至沒有打亂

因此兩點間的路徑長可以直接得到

然後再利用前綴和與sparse table或其他結構取得區間最大值

單次計算路徑長O(1)，(以sparse table為例)建表O(n log n)，單筆查詢O(1)

總體O(n log n+q)

子題三：n <= 200

這個子題對於算路徑長的要求較為寬鬆

可以用弗洛依德(Floyd-Warshall)演算法或其他圖的最短路演算法來求兩點間路徑長

並且查詢不必用資結處理，可以暴力算區間和與區間最大值

(以弗洛依德為例)建表任兩點間路徑長O(n^3)，單筆查詢O(n)

總體O(n^3+nq)

子題一：q <= 10

其實沒那麼好做，這個子題主要是不用解決查詢而已

但為了要快速的在圖上取得兩點間的距離，因此需要LCA進行計算

若不了解可以去熟悉最近共同祖先(LCA)

他可以事先建表(複雜度O(n log n))，然後利用單次O(log n)的複雜度計算樹上兩點距離

計算所有 i 與 i+1 的路徑長O(n log n)，單筆查詢允許O(n)

總體O(n log n+nq)

子題四:無其他限制

結合subtask 1中LCA與subtask 2中的資結

計算所有 i 與 i+1 的路徑長O(n log n), (以sparse table為例)建表O(n log n), 單筆查詢O(1)

總體O(n log n+q)

pG 香彤愛薩梅

出題者 : becaido

實體賽 First AC :

線上賽 First AC :

子題一：n, q <= 300

對於每個詢問，暴力 dfs 算次數，然後枚舉 k，$O(n)$ 檢查它可不可行

複雜度：$O((n+q)n^2)$

子題二：n, q <= 2000

算完每個數字的出現次數後，可以直接算最大公因數 g，g 的因數數量即為答案

複雜度：$O((n+q)n)$

子題三：q = 0

只要算初始節點的所有答案，可以用一個 dfs，每次合併兩個子樹可以把小的併到大的 (啟發式合併)，gcd 用線段樹維護

複雜度：$O(n \log^2(n))$

子題四：無其他限制

如果我們給 x 一個 random 的數字 h[x]，那編號 i 的節點的新權重就是 h[a[i]]，每次詢問以 p 為根的子樹可不可以分成每組 k 個時，檢查 p 子樹的權重和是否為 k 的倍數，當 k>=2，不能成功分組權重和卻是 k 的倍數的機率 <= 1/2，如果用不同 random 值做 t 次機率就 ≤ $1/(2^t)$

於是可以把樹壓平，用 BIT 維護權重和，每次詢問時看 p 對應的區間，求出權重和後可以算他與 sz[p] 的 gcd，做 t 次後取最小的那個，t 可以取 40~50

複雜度：O((n+q) t log(n))

pH 章程和漢堡

出題者 : becaido

實體賽 First AC :

線上賽 First AC :

子題一：q <= 100

暴力 sort，複雜度O(nmq log(nm))

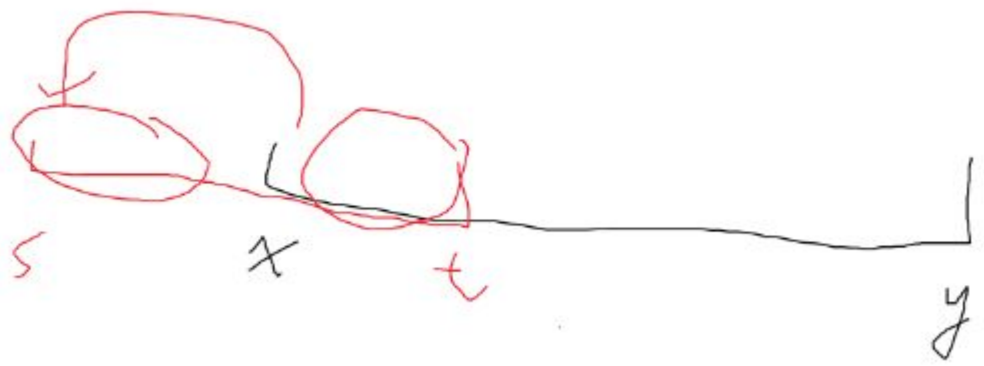
子題二：n=1, 數字皆相異

對 a 的值域開一顆持久化線段樹，複雜度O((m+q) log m)

子題三：n=1

按照 (a[i], w[i]) 排序，用這個順序建一棵持久化線段樹，詢問 [l, r] 的第 x~y 小，可以拆開來算，令第 x 小的是 num，區間 [l, r] 第 s~t 小的也都是 num，那結果會等於 sum(1, y) - sum(1, t) + sum(1, s+t-x) - sum(1, s-1)

但要注意 [s, t] 可能會覆蓋 [l, r] 此時答案是 sum(1, s+y-x) - sum(1, s-1)



子題四：數字皆相異

每次用四個地方線段樹去算 (d, r) - (u-1, r) - (d, l-1) + (u-1, l-1)
至於建法就是當要建 (i, j) 的線段樹，可以從 (i-1, j) 或 (i, j-1) 的線段樹加 i 或 j 個元素，挑比較少的那個來建

複雜度：O((nm min(n, m)+q) log(nm))

子題五：無其他限制

結合子題三和子題四

時間複雜度：$O((nm \min(n, m)+q) \log(nm))$

空間複雜度：$O(nm \min(n, m) \log(nm))$