

# Codeforces 799 (Div.4) 題解

Koying

2022-06-15

- 本次題解的程式碼都可以在 <https://github.com/Koyingtw/IRC-lecture/20220615> 找到

## CF 799A Marathon

有三個人  $a, b, c, d$  在跑步，給他們目前跑的距離，求有幾人的距離比  $a$  遠

- 直接做就好
- 參考程式：PA.cpp

## CF 799B All Distinct

Sho 得到一個長度為  $n$  的序列，他會進行多次操作，每次操作會將序列中的兩個數字刪除，直到沒有任兩個相同的數字為止

求最後序列最長可以多長

( $t \leq 10^3, n \leq 50$ )

- 每種數字最多就只會剩下一個，因此最終目標是要讓剩下一個的數量最大化
- 若有一個數字  $a$  一開始有奇數個，那每次刪掉兩個，最後就會剩下一個
- 若有兩個數字  $a, b$  一開始都有偶數個，那我們只要先將  $a, b$  都  $-1$ ，那就可以將這兩個轉為奇數
- 最終答案就是一開始奇數的數量加上可以從偶數轉成奇數的數量
- 參考程式：PB.cpp

## CF 799C Where's the Bishop?

給定一西洋棋中的主教可以走到的格子（原先在的格子也算），求出原先的格子是哪一格

- 可以發現到，兩條對角線會在原點交會，形成 X 狀
- 若  $(x, y), (x + 1, y + 1), (x + 1, y - 1), (x - 1, y - 1), (x - 1, y + 1)$  都可走的話，那麼  $x, y$  就是原點
- 參考程式：PC.cpp

## CF 799D The Clock

有一個 24 小時制的時鐘，Victor 每  $x$  分鐘會看一次時鐘，問 Victor 在同樣的時間看不超過一次的情況下，會看到幾個回文的時間

- 可以證明由於看不超過 1 次，因此最多就只會看  $24 \times 60$  次
- 按照題目實作即可，時鐘進位的算法請參考下面的程式碼
- 輸入可用 stringstream 來快速處理
- Time complexity:  $\mathcal{O}(24 \times 60)$
- 參考程式：PD.cpp

## 時鐘運行範例

```
int h, m;
void clock_running(int x) {
    m += x;
    h += m / 60;
    m %= 60;
    h %= 24;
}
```

## 回文判斷範例

```
bool ispal(int h, int m) {
    string a = "";
    string b = "";
    a += (h < 10 ? "0": "") + to_string(h) + (m < 10 ? "0": "") + to_string(m);
    b = a;
    reverse(all(b));
    return a == b;
}
```

## CF 799E Binary Deque

有一個長度為  $n$  的 01 字串，你可以從頭或從尾移除數個元素，求最少移除幾個字元能使剩下的字元恰有  $s$  個 1  
( $n, s \leq 2 \cdot 10^5$ )

- 可以觀察到，左界越往右則右界也只會越往右，具有單調性
- 枚舉左界或是右界，利用前綴和差分二分搜出右界或是左界
- Time complexity:  $\mathcal{O}(n \log n)$
- 參考程式: PE.cpp



## CF 799F 3SUM

給一個長度為  $n$  的序列，求是否有可能任取三個元素使得其和的尾數為 3  
( $n \leq 2 \cdot 10^5$ )

- $\mathcal{O}(n^3)$  的作法顯然不可行
- 可以觀察到，三個元素和的尾數是否是 3 只跟三個元素的尾數和有關
- 紀錄每個尾數的數量
- 枚舉三個元素的尾數
- Time complexity:  $\mathcal{O}(n \log n)$
- 參考程式: PF.cpp

## CF 799G 2^Sort

有一個數列  $a$ ，你可以任選一個長度為  $k + 1$  的區間  $[i, i + k]$ ，並進行一次操作，使得區間內的值變為  $2^0 \cdot a_i, 2^1 \cdot a_{i+1}, \dots, 2^k \cdot a_{i+k}$   
求有幾種個區間經過操作之後其區間內的元素呈現嚴格遞增  
( $n \leq 2 \cdot 10^5$ )

- 若  $[a_i, a_{i+1}]$  合法，那麼  $a_i < a_{i+1} \cdot 2$
- 若  $[l, r]$  之間都合法，那麼其中最多包含  $r - l + 1 - k$  個長度為  $k + 1$  的區間
- 找出所有合法的連續區間，即可找到答案
- Time complexity:  $\mathcal{O}(n)$
- 參考程式: PG.cpp

## Codeforces 799H Gambling

Marian 在賭場賭博，有一個遊戲每次會隨機跳一個數字出來，若猜對獎金加倍，猜錯則是獎金折半

Marian 能夠知道接下來每一局遊戲都會出現什麼數字  $x_i$ ，你可以指定一個數字  $a$  使得 Marian 接下來都猜測  $a$ ，以及一個區間  $[l, r]$  代表 Marian 會連續玩  $[l, r]$  的遊戲求 Marian 最多可以獲得多多獎金，他一開始有一塊錢  
( $n \leq 2 \cdot 10^5$ )

- 可以將題目所求轉換成贏得該局則  $+1$ ，輸則  $-1$ ，求最大獎金
- 假設我們要猜的是  $a$ ，那麼對於每次遊戲出現的數字  $x_i$ ，若  $x_i = a$ ，則我們將該局結果  $r_{a,i}$  設為  $+1$ ，反之設為  $-1$ ，那麼猜  $a$  的最大獎金就是  $r_a$  的最大區間和
- 最大區間和算法可由 (當前前墜和 - 走過最小的前墜和) 得出

- 枚舉  $a$ ，找到最大的答案
- 但是  $a \leq 10^9$ ，光是枚舉  $a$  就超時了，因此我們先對他做離散化，讓  $a < n$
- 若每次尋找最大前墜和的過程都是  $\mathcal{O}(n)$ ，那麼整體複雜度會來到  $\mathcal{O}(n^2)$
- 觀察一下  $r_a$  會發現，他其實就是幾個  $+1$  中間插入很多個  $-1$  組成的，再加上最好的邊界一定落在  $r_{a,i} = +1$  的位置，所以我們只需要紀錄那些點是  $+1$  即可
- 每次計算最大前墜和的複雜度變為  $\mathcal{O}(cnt_a)$
- 最終的複雜度就會是  $\mathcal{O}(\sum cnt_a) = \mathcal{O}(n)$  加上離散化的  $\mathcal{O}(n \log n) = \mathcal{O}(n \log n)$