

# 建中校內第六次模擬賽題解

hansonyu123, Yihda Yol

# 房屋出租問題 Subtask 1-2

- 暴力枚舉，複雜度  $O(C_n) = O\left(\frac{4^n}{n\sqrt{n}}\right)$

# 房屋出租問題

- DP-able
- 令  $dp(i, j)$  是三角化  $[i, j]$  所需要的經費（因為是環狀，有可能  $i > j$ ）
- $dp(i, j) = \min_{k \in (i, j)} dp(i, k) + dp(k, j) + \overline{P_i P_j}$
- $O(n^3)$
- 記錄轉移來源，最後用遞迴回溯回去就可以了（記得不要輸出原本凸多邊形的邊）

# 偏序關係 Subtask 1

- 先將遞移律顯現在圖中
  - 如果 $a \rightarrow b$ ， $b \rightarrow c$ ，那麼 $a \rightarrow c$
  - 貌似Floyd-Warshall
- 先利用Floyd-Warshall將圖建好，將有雙向邊的點縮成一個點（並查集），剩餘點數就是最大相異元素數。可以發現只要把入度為0的點設為最小後拔掉，再繼續找最小就是好的排序法
- 複雜度 $O(n^3 + n\alpha(n) + n^2) = O(n^3)$

# 偏序關係

- 前面縮點的方法相當於是在找SCC，縮完點後的排序相當於是找拓撲排序
- 聽說兩個都可以線性時間做完... ..
  - 那我還要講什麼

# 人生低潮 Subtask 1

- 直接找最小值就好
  - 複雜度 $O(NM)$ ，TLE 13分
- 另解：每次都走到在他隔壁而且比他小的那格，一直走下去，直到有一天會走不下去。該格即為所求。
  - 複雜度 $O(NM)$ ，不過在隨機測資中表現會很好

# 人生低潮 Subtask 2

- 按照剛剛一直往下走的想法，可以想出  $O(\log M)$  的作法。
- 隨便戳一格。如果左邊比較小，代表左邊一定有一個低潮。如果右邊比較小，代表右邊一定有個低潮。
- 用這種方法就可以對序列進行類似二分搜的事
  - 對於  $[l, r]$ ，用  $m$  將區間分成  $[l, m-1][m+1, r-1]$ 。如果  $m$  比  $l-1, r+1$  還小，就用上面的方法選左右。如果  $l-1$  比較小，就選左邊。

## 人生低潮 Subtask 3

- 看起來要 $O(N)$ 才會過。我們試著用 $O(N)$ 的時間將 $N \times N$ 的問題變為 $\left(\frac{N}{2}\right) \times \left(\frac{N}{2}\right)$ 的問題。
- 仿照Subtask 2的方法，將 $N \times N$ 切成四片，想辦法找出哪塊一定有低潮。
- 用 $O(N)$ 的時間找出 $N \times N$ 的兩條中折線的最小值，就可仿前面的方法找出一定有低潮的某塊。
- $T(N) = T\left(\frac{N}{2}\right) + O(N)$ ，故 $T(N) = O(N)$



# 人生低潮

- 結合前面兩個subtask的方法
  - 先用Subtask 2二分搜的方法將M變成和N差不多大： $O\left(N \log \frac{M}{N}\right)$
  - 再用Subtask 3的作法解決掉： $O(N)$
  - 總複雜度 $O\left(N \left(1 + \log \frac{M}{N}\right)\right)$
- 另解：不妨設 $N < M$ ，就把 $N \times M$ 對半切成兩塊 $N \times (M/2)$ ，用前面的方法挑出一塊候選人。複雜度同樣是 $O\left(N \left(1 + \log \frac{M}{N}\right)\right)$ 。

# 人生低潮

- $O(N \log M)$  也會過呵呵。

# 磁石陣

- Subtask 1 : 枚舉所有放入順序
- Subtask 2 : DP , 狀態數 $O(2^n)$  , 總複雜度 $O(n^2 2^n)$  。

## 磁石陣 Subtask 3

- 觀察：對於最後一個放入磁石陣的磁石來說，不論前面怎麼亂放，最後一個磁石所受的斥力都一樣。
- 可以發現讓最後一個磁石是所受斥力最小的是一個好的greedy策略。決定好把磁石放在哪後就將他的磁場消掉，再從剩餘磁石中找所受斥力最小的那顆磁石。
- 複雜度 $O(n^2)$ ，TLE 67分。

# 磁石陣

- 需要將「消去磁場範圍」以及「尋找最小值」兩個操作加速
- 「消去磁場範圍」可以用常數個區間修改替代，「尋找最小值」即是區間最小值的特例。利用線段樹以及懶人標即可在每次  $O(\log n)$  的時間完成任務。
- 複雜度  $O(n \log n)$ ，AC。

# 魔法陣添屎 Subtask 1

- 每次都直接找凸包+極角排序。複雜度  $O(N^2 \log N)$ 。

# 魔法陣添屎 Subtask 2

- 用set維護極角排序
- 每次插入時找到他在set的位置，看看他、前一個和後一個人是否是凸角。確定要放入後再將被包在凸邊形內部的人移去。
- 攻擊時直接用lower\_bound往後數k個。
- 每個人最多插入移除一次，故總複雜度  $O(N(\log N + \min(N, k)))$ ，TLE 58分。

# 魔法陣添屎

- **Set**不支援快速往後找k個人的功能，所以只好自己寫一個自平衡二元樹
- **Treap**支援所需的所有功能
- 複雜度 $O(N \log N)$ ，AC
- 實作有許多小細節要注意（比如說浮點運算精度不夠需要用整數運算，角度剛好相等或相差 $180^\circ$ 時的特殊處理），不過這題的測資沒有刻意卡這些。



# 同步 Subtask 1

- 直接用快速幂暴搜，複雜度 $O(NQ \log P)$

## 同步 Subtask 3

- 不難發現在這個subtask中1只跟1同步，p-1只跟p-1同步
- 問題等價於兩種字元的LCP問題，用一般的suffix array以及RMQ問題的作法（因為Q偏大，這裡採用sparse table）可以得到時間複雜度 $O(N \log N + Q \log N)$ 、空間複雜度 $O(N \log N)$ 的方法。
- WA 37分

# 同步

- 會不會整題其實都是LCP呢？
- 由費小知 $a^{\frac{p-1}{2}} = \pm 1$ ，所以a跟b同步的充要條件是 $a^{\frac{p-1}{2}} = b^{\frac{p-1}{2}}$ ，問題轉化為兩種字元的LCP問題。
- 複雜度  $O(N(\log p + \log N) + Q \log N)$  ， AC 。