演算法與資料結構（三）

大叫

目錄

預處理 Preprocessing

預處理 Preprocessing

- 預處理是什麼？
- 在演算法中，先花一點時間對資料進行處理，以利後續的操作
- 現實的例子？

預處理 Preprocessing

- CSES Static Range Sum Queries
- 給 n, q 和長度為 n 的數列，接下來詢問 q 次某個區間 [l, r] 內的數字和
- $1 \le n,q \le 2e5$

- 最暴力的做法：每次詢問都重新掃過 l~r 來計算區間和，總複雜度 O(nq)
- 可以預處理什麼，使詢問的複雜度下降？
- 前綴和

預處理 Preprocessing

- 給 n, m, q 和 n*m 個數組成的矩陣，接下來詢問 q 次由某個 [x1, y1] 與 [x2, y2] 組成的矩形，其數字和為多少
- $1 \le n,m \le 2000, q \le 2000$

- 上一題的延伸
- 怎麼對二維的目標取前綴和，使每次詢問都是 O(1) 嗎？

預處理 Preprocessing

- 你有 10 種幣值的鈔票各一張，和 q 次詢問能否剛好湊出 x 元。幣值分別為 1, 5, 10, 50, 100, 500, 1000, 5000, 10000, 50000 元
- $1 \le q \le 10^7$, $1 \le x \le 100000$

- 每次位元枚舉可用 $O(2^{10})$ 的複雜度得到所有可能湊出的總額，複雜度 $O(q*2^{10})$ 絕對爆開
- 如何用預處理優化?

雙指針 Two Pointers

雙指針 Two Pointers

- 跟枚舉有點像
- 利用單調性，來省略不必要的枚舉
- 實作上真的就是兩個指針在跑

雙指針 Two Pointers

- 示範題
- 有兩個長度為n且遞增的序列 A,B ，你可以從中各選擇一個數，問有幾種選法會使選擇的兩個數的和大於 m
- 1≤n≤2e5

- 因為 n 的範圍很大，無法用 $O(n^2)$ 的方式枚舉答案

雙指針 Two Pointers

- CSES 1640 Sum of Two Values
- 有個長度為n的序列a和數字 m，求兩個不同的位置 i,j 滿足 a[i]+a[j]=m
- $1 \le n \le 2e5$, $a[i] \le 10^9$

雙指針 Two Pointers

- CSES 1641 Sum of Three Values
- 有個長度為n的序列a和數字 m，求三個不同的位置 i,j,k 滿足 a[i]+a[j]+a[k]=m
- 1≤n≤5000

雙指針 Two Pointers

- CF edu Two Pointers: Number of Segments with Small Sum
- 給一個長度為 n 的序列 a 和數字s，並詢問 a 中有幾個區間滿足區間內元素的和小於 s
- 1 ≤ n ≤ 2e5, 1 ≤ a ≤ 1e9, 1≤s≤1e18

二分搜 Binary Search

二分搜 Binary Search

- 每個人的二分搜都有自己的寫法，平時只要用自己最熟悉的方式就好了，這邊講二分搜時主要用我習慣的寫法，用[L,R]維護「答案可能的區間」

二分搜 Binary Search

- 示範題
- 現在要在1~100之間猜一個數字，若猜錯了則會告訴你答案更大或更小，有什麼策略可以保證在最糟的狀況下，猜到答案的次數最小?

二分搜 Binary Search

- 剛剛的例子已經知道，二分搜的核心概念就在於「二分」
- 如何將問題給一般化：有一個0/1組成的序列，任意0都在任意1的左邊（如00011），用二分搜的方式找出從左往右第一個出現的第一個1？
- 用剛剛的題目做個舉例

二分搜 Binary Search

- 雖然二分搜本身的概念不難，但細節和寫法都很多，還容易出BUG
- 來看看二分搜都有哪些BUG
  - 1.沒有單調性(忘記排序、題目本來就不符合)
  - 2.初始左右界設錯，導致L~R太小答案不再搜尋範圍中，或L太小、R太大，得到mid後卻在計算過程overflow
  - 3.L+R時overflow
  - 4.負數運算
- 解決方法則是
  - 1.~~下次注意~~
  - 2.思考極端狀況下L跟R的範圍，若是怕範圍太大計算數值時overflow，則L跟R可以用數學公式算，或者就在計算前先特判會不會overflow
  - 3和4： 將寫法從(L+R)/2改為L+(R-L)/2
- 為什麼寫L+(R-L)/2可以解決負數運算的問題?

二分搜 Binary Search

- 思考細節
- 1.當前區間大小為 2 時，mid 會指向左邊還右邊的元素
- 2.當L跟R移動時，它們跟 mid 的關係
- 3.當跳出迴圈時，答案的位置跟 L 或 R 的關係

二分搜 Binary Search

- c++內建的二分搜工具：
- binary_search(it_L, it_R, val)：回傳一個 bool，代表在 [it_L,it_R) 中是否有找到 val
- lower_bound(it_L, it_R, val)：回傳一個 iterator，指向 [it_L,it_R) 中第一個不小於 val 的位置
- upper_bound(it_L, it_R, val) 回傳一個 iterator，指向 [it_L,it_R) 中第一個大於val的位置

二分搜 Binary Search

- 使用內建工具的好處：
- 比自己寫快
- 只要使用方式正確就不會避免出錯

二分搜 Binary Search

- CF edu Two Pointers: Number of Segments with Small Sum
- 給一個長度為n的序列a和數字s，並詢問a中有幾個區間滿足區間和不大於s
- 1≤n≤2e5，1≤a的任意元素≤1e9，1≤s≤1e18
- 哈哈這個題序是不是有點眼熟

對答案二分搜

- 對答案二分搜，顧名思義就是對答案二分搜
- 使用時機：發現答案符合 0/1 序列的性質，且不會TLE時

對答案二分搜

- CSES 1620 - Factory Machines
- 有 n 台機器，第 i 台機器可以用 ki 的時間製造一個產品，問製造 t 個產品最少需要多久
- $1 \le n \le 2e5$, $1 \le t \le 109$, $1 \le ki \le 1e9$

對浮點數二分搜

- 對浮點數二分搜，顧名思義就是對浮點數二分搜

```
int main() {
    double L = 左界;
    double R = 右界;
    for(int i=0;i<200;i++) {
        double mig = (L+R)/2;
        if(ok(mid)) {
            R = mid;
        }
        else {
            L = mid;
        }
    }
}
```

對浮點數二分搜

- ITMO Academy: pilot course » Binary Search » Step 2 » B. Ropes
- 給n條長度不一定相同的繩子與數字k，問如果用這n條繩子剪出k條長度一樣的繩子，這些繩子的最大長度為多少(誤差度為-1e6)
- $1 \le n, k \le 1e4$ ，$1 \le$ 繩子的長度 $\le 10^7$

- 要搜什麼？

QA

- 例如：
- 對前面的內容有哪裡想問？
- 其他...

內容回顧

- 預處理 Preprocessing
- 雙指針 Two Pointers
- 二分搜 binary search

謝謝各位

大叫