

พออ่านชื่อแล้ว จะชวนให้คิดว่าเป็น binary search หลายๆอันพร้อมกัน

(ย่อหน้านี้อ่านผ่านๆไปก่อนก็ได้) **Paralell binary search** คือ การที่โจทย์มี Q คำถาม แล้วแต่ละคำถามเราต้องใช้ binary search แก่ แต่ทว่าการทำ binary search ของแต่ละคำถามมันซ้ำมากเช่น $O(n \lg n)$ ถึงจะแก้ได้ ทำให้เวลารวมเป็น $O(qn \lg n)$ ซึ่งช้าเกินไป

ลองดูโจทย์ข้อ meteor ที่เว็บ codeforces ยกตัวอย่างมากันครับ

มีรัฐอยู่ N รัฐ และมีพื้นที่อยู่ M ช่อง พื้นที่แต่ละช่องจะเป็นของรัฐใดรัฐหนึ่ง
ที่วันนี้เราสนใจจะมีทั้งหมด Q วัน แต่วันจะมีอุกกาบาตตกลงมา
ในวันที่ i จะมีอุกกาบาตจำนวนหนึ่งตกลงมาในพื้นที่หมายเลข $[L_i, R_i]$ (L_i ถึง R_i)
รัฐที่ i อยากเก็บตัวอย่างอุกกาบาตในพื้นที่ของตนอย่างน้อย $req[i]$
อยากรู้ว่า รัฐแต่ละรัฐต้องรออย่างน้อยกี่วันถึงจะเก็บตัวอย่างอุกกาบาตได้ตามต้องการ
($1 \leq N, M \leq 10^5$)

(โจทย์อาจจะแปลมาไม่ครบ แต่เอาใจความสำคัญหลักๆพอครับ)

Original : There are N member states and M sectors. Each sector is owned by a member state. There are Q queries, each of which denote the amount of meteor shower in a $[L, R]$ range of sectors on that day. The i^{th} member state wants to collect $reqd[i]$ meteors over all its sectors. For every member state, what is the minimum number of days it would have to wait to collect atleast the required amount of meteors?

ถ้าทำตรงๆเลยก็ให้หาทีละรัฐตั้งแต่ $1, \dots, N$ โดยทำ binary search วันของรัฐนั้นๆ(อาจจะต้องใช้ fenwick tree ช่วย)

ตอนเรา Binary Search สิ่งที่เราทำคือ

1. หา mid
2. ไล่อัปเดต fenwick เป็นช่วงตั้งแต่วันที่ 1 ถึงวันที่ mid
3. หาผลรวมอุกกาบาตของทุกช่องที่เป็นของรัฐที่เรากำลังทำ แล้วดูว่าได้หรือไม่

ซึ่งจะเป็น $O(N * (\log Q * Q * \log M) + M * \log Q * \log M)$ ซึ่งไม่ทันแน่นอน...

เราจะมาทำ binary search ทุกรัฐพร้อมๆกันไปเลย!

สังเกตว่า ทุกรัฐที่เราทำมันต้องมี

“2. ไล่อัปเดต fenwick เป็นช่วงตั้งแต่วันที่ 1 ถึงวันที่ mid”

Paralell Binary Search

1. ตอนเริ่มต้น ให้ l กับ r ของทุกรัฐเป็น 1 กับ Q
2. ให้เราดูว่าแต่ละรัฐมี mid เป็นวันไหนแล้วยัดลง vector[วันนั้น] (เราจะเอา vector นี้ไปใช้ในข้อ 3)

3.ให้เราวนไล่อัปเดต fenwick เป็นช่วง ตั้งแต่วันที่ 1 ถึงวันที่ Q

-แต่คราวนี้ หลังจากเราอัปเดต fenwick ของวันๆหนึ่งเสร็จ ให้เราดูว่ามีรัฐไหนที่มี mid เป็นวันนี้บ้าง ให้ดูจาก vector ที่เราทำไว้

-แล้วให้รัฐนั้นทำการวนไล่หาผลรวมออกกบาทของทุกช่องที่เป็นของรัฐ แล้วดูว่าได้หรือไม่ได้

-ให้เราอัปเดต l,r ของรัฐนั้น

4.วนกลับไปทำข้อ 2 จนกว่าจะ binary search เสร็จทุกรัฐ(แต่ละรัฐจะ binary search ไม่เกิน $\log Q$ ครั้ง ดังนั้นเราจะวนประมาณแค่ $\log Q$ ครั้ง) อย่าลืม clear vector กับ fenwick ด้วย

Pseudo Code

```
for all logQ steps:
    clear range tree and linked list check
    for all member states i:
        if L[i] != R[i]:
            mid = (L[i] + R[i]) / 2
            insert i in check[mid]
    for all queries q:
        apply(q)
        for all member states m in check[q]:
            if m has requirements fulfilled:
                R[m] = q
            else:
                L[m] = q + 1
```

(ตรง linked list ให้เราใช้ vector แทน)

คราวนี้ จะเหลือแค่ $O(\text{[การทำข้อ 2-4 ทั้งหมด } \log Q \text{ ครั้ง]} + \text{[การไล่เช็คออกกบาทของทุกรัฐ ทั้งหมด } \log Q \text{ ครั้ง]})$

$= O(Q \cdot \log M \cdot \log Q + N \cdot \log M \cdot \log Q)$

ซึ่งทันเวลาแล้ว!