Bài 2. Truy vấn tổng tĩnh

🧠 Mô tả bài toán

> Cho mảng số nguyên `A[]` gồm `N` phần tử. Có `Q` truy vấn, mỗi truy vấn gồm hai chỉ số `L` và `R` (0-indexed). Với mỗi truy vấn, bạn cần tính tổng các phần tử từ `A[L]` đến `A[R]`.

🧾 Input Format

- Dòng 1: Hai số nguyên `N` và `Q` — số phần tử và số lượng truy vấn.
- Dòng 2: `N` số nguyên — các phần tử của mảng `A[]`.
- `Q` dòng tiếp theo: mỗi dòng gồm hai số nguyên `L` và `R` — biểu diễn một truy vấn.

📌 Ràng buộc

- $1 \leq N, Q \leq 10^6$
- $1 \leq A[i] \leq 10^9$
- $0 \leq L \leq R \leq N$ - 1

🔢 Output Format

- Với mỗi truy vấn, in ra tổng các phần tử từ chỉ số `L` đến `R`, mỗi kết quả trên một dòng.

📥 Ví dụ minh hoạ

Input:

```
11 3
6 5 9 7 7 6 7 7 5 9 7
1 9
1 10
0 8
```

Output:

```
62
69
59
```

Giải thích:

- Truy vấn 1: Tổng từ `A[1]` đến `A[9]` là `5+9+7+7+6+7+7+5+9 = 62`
- Truy vấn 2: Tổng từ `A[1]` đến `A[10]` là `62 + 7 = 69`
- Truy vấn 3: Tổng từ `A[0]` đến `A[8]` là `6+5+9+7+7+6+7+7+5 = 59`

💡 Ý tưởng thuật toán

Để trả lời nhanh nhiều truy vấn tổng đoạn, ta dùng **mảng cộng dồn một chiều**.

Gọi:

- `prefix[i]` là tổng các phần tử từ `A[0]` đến `A[i]`.

Khi đó:

- Tổng từ `L` đến `R` được tính nhanh bằng:
  - `prefix[R] - prefix[L - 1]` nếu `L > 0`
  - `prefix[R]` nếu `L == 0`

🧮 Công thức mảng cộng dồn

- `prefix[0] = A[0]`
- `prefix[i] = prefix[i - 1] + A[i]` với `i ≥ 1`

⏱️ Độ phức tạp

- **Thời gian tiền xử lý:** O(N)
- **Thời gian mỗi truy vấn:** O(1)
- **Tổng thời gian:** O(N+Q)

→ Phù hợp với ràng buộc lớn (tối đa 10^6).