Trả lời Q truy vấn trong đó mỗi truy vấn có dạng:

$$\sum_{L \le i,j \le R} (A_i \ mod \ A_j)$$

Subtask 1

Lời giải là thực hiện theo những gì đề yêu cầu.

Độ phức tạp: $O(Q \times N^2)$.

Subtask 2

Tại đây chúng ta sẽ biến đổi công thức ban đầu:

$$\sum_{L \le i,j \le R} (A_i \ mod \ A_j)$$

Tương đương:

$$cnt[A_i] \times cnt[A_j] \times (A_i \ mod \ A_j)$$

Hay phát biểu lại thành dạng chữ là:

Ta có thể thấy khi một cặp (A_i, A_j) được xét thì kết quả sẽ tăng một lượng là:

Số lần xuất hiện của A_i trong đoạn $[L; R]$ × Số lần xuất hiện của A_j trong đoạn $[L; R]$ × $(A_i \ mod \ A_j)$

Vậy chúng ta sẽ dùng mảng cộng dồn và mảng đánh dấu để tính nhanh số lần xuất hiện của A_i và A_j trong đoạn $[L; R]$ rồi cố định i và j (được hiểu là giá trị của A) để duyệt $(A_i \ mod \ A_j)$.

Độ phức tạp: $O(Q \times V^2)$ trong đó V là giá trị tối đa của A.

Subtask 3

Vì subtask này có sự bất thường là N đã giảm xuống còn 10^3 khi này chúng ta sẽ cùng tìm một cách giải với độ phức tạp xoay quanh $O(N^2)$ và sẽ xem cách đó có hợp lệ hay không.

Ta sẽ khai triển công thức để thấy được cách để có thể thấy được cặp (i, j):

$$\sum_{L \le i,j \le R} (A_i \ mod \ A_j)$$

Hay sẽ chính là:

$$\sum_{L \le i \le R} \sum_{L \le j \le R} (A_i \ mod \ A_j)$$

Tại đây nếu như coi i và j là tọa độ của một hình chữ nhật lớn thì có thể thấy kết quả tính nhanh của công thức trên sẽ chính là mảng cộng dồn 2 chiều mọi của mọi cặp (i, j) có thể.

Gọi mảng B với $B_{i,j} = A_i \ mod \ A_j$.

Khi đó truy vấn $[L; R]$ sẽ chính là tổng của hình chữ nhật $pref[L; R][L; R]$ với $pref$ chính là mảng cộng dồn của B (góc trái trên là (L, L), góc phải dưới là (R, R)).

Subtask 4

Tại đây một lần nữa đã có sự giảm bất thường của A_i tối đa hiện tại chỉ 10^3 vậy chúng ta sẽ xem có cách nào tối ưu lời giải subtask 2 hay không.

$$cnt[A_i] \times cnt[A_j] \times \left(A_i \ mod \ A_j\right)$$

Khai triển công thức hiện tại nếu nhìn kĩ chúng ta sẽ thấy một điều là:

Có thể tối ưu thêm bằng cách bỏ đi việc duyệt A_j mà chúng ta sẽ tính luôn sẵn kết quả $A_i \ mod \ A_j$. Cách này hợp lệ vì tối đa cũng chỉ có khoảng $10^3 \times 10^3$ kết quả khác nhau của cặp $A_i \ mod \ A_j$.

Tức thay vì đếm số lượng cặp rồi nhân cho nhau (là chúng ta đang lấy kết quả rồi nhân cho số cặp tương ứng) thì chúng ta sẽ tính xem với A_j hiện tại (tức chúng ta cố định A_j rồi kiểm tra với khoảng $[L; R]$) thì những phần tử A_i trong $[L; R]$ tạo được tổng là bao nhiêu (phần này có thể tính nhanh bằng công thức khai triển prefix sum bên dưới).

$$pref[i][j] = pref[i-1][j] + (A_i \ mod \ j) = \sum_{k=1}^{i} (A_k \ mod \ j)$$

Với j là số đang mod hiện tại (vậy tối đa j chỉ có thể là 10^3) và i chính là vị trí hiện tại trong mảng.

Khi đó công thức sẽ chỉ còn:

$$cnt[j] \times (pref[R][j] - pref[L-1][j])$$

Lưu ý: $cnt[A_j]$ bây giờ đã chuyển thành $cnt[j]$ và $cnt[A_i]$ không cần thiết nữa vì đã được nhập vào trong tổng tiền tố $pref$ (ban đầu chúng ta cần quan tâm số lượng vì chúng ta đếm dựa trên số cặp nhưng bây giờ chúng ta sẽ đếm trên khoảng tổng tiền tố $pref$ trong khoảng $[L; R]$ khi đó việc A_i xuất hiện bao nhiêu lần hay $cnt[A_i]$ thì đã được tính trong $pref$).

Còn $cnt[j]$ chính là số lần xuất hiện của j trong khoảng $[L; R]$ giải bằng mảng đánh dấu và mảng cộng dồn.

Độ phức tạp: $O(V^2 + Q \times 1000)$ với V là giá trị tối đa của A.

[Solution mẫu]