SOLUTION 13/12 - S

Ước đặc biệt

Trường hợp 1:

- Duyệt tất cả các số i từ 1 đến N - 1, kiểm tra xem i và i + 1 có phải ước của N không.

Trường hợp 2:

- Duyệt tất cả các số i từ 1 đến $\sqrt{N}$, kiểm tra xem i và i + 1 có phải ước của N không. Lưu í các số > $\sqrt{N}$ không thỏa mãn là ước đặc biệt của N.

Trường hợp 3:

- Duyệt tất cả các số từ 1 đến $N^{\frac{1}{3}}$, kiểm tra xem i và i + 1 có phải ước của N không.
- Với những ước đặc biệt i > $N^{\frac{1}{3}}$, ta thấy N ⋮ i * (i + 1)

=> N = k * i * (i + 1) với k nguyên dương và k < $N^{\frac{1}{3}}$.

Do đó, ta duyệt hết các số k < $N^{\frac{1}{3}}$ là ước của N, rồi kiểm tra xem tồn tại số i thỏa mãn:

i * (i + 1) = N /k => i ≅ $\sqrt{\frac{N}{K}}$. thì i là 1 ước đặc biệt của N.

CNTPAL

Sub 1:

_ Với mỗi truy vấn (u,v), duyệt hết các đoạn (i,j) trong đoạn (u,v), kiểm tra xem nó có phải là xâu đối xứng không bằng cách đảo ngược xâu đó lại, rồi xem xâu đảo ngược có bằng xâu đó không.

_ Độ phức tạp: (Q * N * N * N)

Sub 2:

_ Ta gọi QHĐ dp[i[j] là kiểm tra xem xâu con (i,j) có phải xâu đối xứng ko. Công thức là:

dp[i][j] = dp[i + 1] [j - 1] (nếu s[i] == s[j])

_ Sau đó, với mỗi truy vấn (u,v), ta duyệt hết các đoạn (i,j) trong đoạn (u,v), kiểm tra xem nó có phải xâu đối xứng không (dp[i][j] có bằng 1 không?)

_ Độ phức tạp: (Q * N * N)

Sub 3:

_ Với mỗi số i, ta sẽ biết được những giá trị j nào để xâu con (i,j) đối xứng (hay dp[i][j] = 1). Ta tống hết các giá trị j này vào vector thứ i.

_ Với mỗi truy vấn (u,v), ta duyệt hết các giá trị i từ u -> v, sau đó đếm các giá trị j trong vector thứ i mà u ≤ j ≤ v. Hoàn toàn có thể sử dụng chặt nhị phân để tính.

_ Độ phức tạp: (Q * N * log(N))

Sub 4:

_ Ta gọi f[i][j] là số giá trị k từ i -> j thỏa mãn: dp[i][k] là xâu đối xứng. Công thức:

f[i][j] = dp[i][i] + dp[i][i + 1] + … + dp[i][j].

_ Ta gọi g[i][j] là số cặp giá trị (u,v) nằm trong đoạn (i,j) thỏa mãn: dp[u][v] là xâu đối xứng. Công thức:

g[i][j] = f[i][j] + f[i + 1][j] + … + f[j][j]

_ Khi đó với mỗi truy vấn (x,y), kết quả đơn giản là g[x][y].

_ Độ phức tạp: O(N * N + Q)

Tính diện tích

Trường hợp 1:

- Đây là bài toán tìm diện tích hình chữ nhật phủ với $N \leq 1000$.
- Độ phức tạp (N^2)

Trường hợp 2:

- Do tọa độ ≤ 1000, ta có thể duyệt từng điểm (i,j) để kiểm tra xem nó có bị phủ hay không. Có 2 cách:
  + Dùng mảng cộng dồn 2D.
  + Mỗi hcn (x,y,z,t) được biểu diễn bởi 2 đoạn (x,y,t) và (z,y,t). Sort theo hoành độ tăng dần, duy trì các đoạn theo điểm (i,j) mình đang xét theo hoành độ, cuối cùng dùng 1 segment tree để kiểm soát tung độ xem điểm (i,j) có bị phủ không.
- Nếu điểm (i,j) bị phủ, chỉ cần cộng $(i * j)^k$ vào kết quả.

Trường hợp 3:

- Đây là biến thể của bài toán tìm diện tích phủ với $N \leq 10^5$.
- Để ý 1 hình chữ nhật(x,y,z,t) trong bài này có diện tích phủ là:
  $(x^k + (x + 1)^k + \ldots + z^k) * (y^k + (y + 1)^k + \ldots + t^k)$
  $= (f[z] - f[x - 1]) * (f[t] - f[y - 1])$ với $f[i] = 1^k + 2^k + \ldots + i^k$
  so với diện tích hình chữ nhật (x,y,z,t) có diện tích phủ gốc là: $(z - x) * (t - y)$
  => Cách làm y hệt bài toán tìm diện tích hình chữ nhật phủ cơ bản, hoàn toàn có thể tính trước các f[i] với $i \leq 10^5$, và không phải nén số trong segment tree.
- Độ phức tạp $(N\ Log\ N)$

Trường hợp 4:

- Với bài này tọa độ rất lớn nên ta phải xử lý thêm 2 việc:
+ Nén các tọa độ về cỡ N để có thể sử dụng segment tree.
+ Tính nhanh các giá trị f[i].
- Để tính nhanh f[i]:
+ Với k = 2, ta có: $f[i] = 1^2 + 2^2 + \ldots + i^2 = \frac{i * (i + 1) * (2i + 1)}{6}$
+ Với k = 3, ta có: $f[i] = 1^3 + 2^3 + \ldots + i^3 = \frac{i^2 * (i + 1)^2}{4}$
- Độ phức tạp $(N\ Log\ N)$

Trường hợp 5:

- Ta phải đi tính nhanh $f[i] = 1^k + 2^k + \ldots + i^k$ với $k \leq 20$ và $i \leq 10^9$.
- Đây là bài toán sử dụng kiến thức khá lạ là nội suy Lagrange để giải quyết bài toán trên.

---------HẾT---------