Daily Codeforces – 10/01/2026

Trieu Vi

Problem 1, 2: Expected Beauty of a Palindromic Mosaic

`https://codeforces.com/contest/2175/problem/E2`

Tóm tắt đề bài

Cho một dãy ngẫu nhiên độ dài n, mỗi ký tự được chọn độc lập và đồng đều từ m màu. Gọi f là số đoạn con không rỗng đối xứng (palindrome). Yêu cầu tính giá trị kì vọng:

$$\mathbb{E}[f^2].$$

Lời giải

Gọi $x_{l,r}$ là biến chỉ thị ứng với đoạn con $[l, r]$:

$$x_{l,r} = \begin{cases} 1, & \text{nếu } s[l..r] \text{ là palindrome,} \\ 0, & \text{ngược lại.} \end{cases}$$

Khi đó:

$$f = \sum_{[l,r]} x_{l,r}.$$

Suy ra:

$$\mathbb{E}[f^2] = \mathbb{E}\left[\left(\sum x_{l,r}\right)^2\right] = \mathbb{E}\left[\sum x_{l,r}^2 + 2\sum x_{l_1,r_1}x_{l_2,r_2}\right].$$

Do $x_{l,r}^2 = x_{l,r}$, ta có:

$$\mathbb{E}[f^2] = \mathbb{E}\left[\sum x_{l,r}\right] + 2\,\mathbb{E}\left[\sum x_{l_1,r_1}x_{l_2,r_2}\right].$$

Tính $\mathbb{E}[x_{l,r}]$

Xét một đoạn con có độ dài

$$s = r - l + 1.$$

Để đoạn này là palindrome, ta chỉ cần chọn tự do

$$\left\lceil \frac{s}{2} \right\rceil = \frac{s+1}{2}$$

ký tự đầu; các ký tự còn lại được xác định theo đối xứng.

Do đó:

$$\mathbb{E}[x_{l,r}] = \frac{m^{(s+1)/2}}{m^s}.$$

Tính $\mathbb{E}[x_{l_1,r_1}x_{l_2,r_2}]$

Xét hai đoạn con có độ dài lần lượt là s_1 và s_2.

Trường hợp 1: Hai đoạn đồng tâm Nếu hai đoạn có cùng tâm và cùng tính chẵn/lẻ, thì đoạn dài hơn quyết định hoàn toàn đoạn ngắn hơn. Khi đó:

$$\mathbb{E}[x_{l_1,r_1}x_{l_2,r_2}] = \frac{m^{(\max(s_1,s_2)+1)/2}}{m^{\max(s_1,s_2)}}.$$

Trường hợp 2: Hai đoạn không đồng tâm Khi hai đoạn không đồng tâm, ta chứng minh rằng hai sự kiện "đoạn con là palindrome" là độc lập. Do đó:

$$\mathbb{E}[x_{l_1,r_1}x_{l_2,r_2}] = \frac{m^{(s_1+1)/2}}{m^{s_1}} \cdot \frac{m^{(s_2+1)/2}}{m^{s_2}} = \frac{m^{(s_1+1)/2+(s_2+1)/2}}{m^{s_1+s_2}}.$$

Chứng minh tính độc lập

Nếu hai đoạn không giao nhau thì điều này là hiển nhiên. Giả sử hai đoạn giao nhau, gọi len là độ dài phần giao. Mỗi vị trí trong phần giao có hai điểm đối xứng:

- một theo tâm của đoạn thứ nhất,
- một theo tâm của đoạn thứ hai.

Do hai đoạn không đồng tâm, các cặp điểm đối xứng này luôn khác nhau và không chồng lấp. Vì vậy:

- đoạn thứ nhất có $(s_1 + 1)/2$ ký tự được chọn tự do,
- đoạn thứ hai chỉ cần chọn thêm $(s_2 + 1)/2 - \text{len}$ ký tự.

Suy ra:

$$\mathbb{P}(\text{cả hai là palindrome}) = \frac{m^{(s_1+1)/2+(s_2+1)/2-\text{len}}}{m^{s_1+s_2-\text{len}}} = \frac{m^{(s_1+1)/2+(s_2+1)/2}}{m^{s_1+s_2}}.$$

Tối ưu hoá

Ta tách riêng:

- các cặp đoạn đồng tâm,
- các cặp đoạn không đồng tâm.

Với các cặp không đồng tâm, tổng cần tính có dạng:

$$\sum m^{-(s_1/2+s_2/2)}.$$

Cố định tính chẵn/lẻ của s_1 và s_2, ta chỉ cần duyệt theo:

$$s = s_1 + s_2,$$

và nhân

$$m^{-(s/2-\text{flag})}$$

với số cặp đoạn không đồng tâm có tổng độ dài bằng s, trong đó flag $= 1$ khi và chỉ khi cả hai đoạn đều có độ dài lẻ. Nhờ đó, toàn bộ thuật toán đạt độ phức tạp:

$$O(n).$$

Problem 3: Secret Message

`https://codeforces.com/contest/2175/problem/F`

Tóm tắt đề bài

Bạn được cho một **đồ thị vô hướng** gồm n đỉnh và m cạnh, mỗi cạnh có trọng số. Nhiệm vụ là chọn **đúng** $n-1$ **cạnh** sao cho:

- Tổng trọng số là **nhỏ nhất**
- Tập cạnh được chọn **không tạo thành một cây**

Nếu **không tồn tại** cách chọn như vậy, hãy in ra -1.

Lời giải

Ta sắp xếp các cạnh theo thứ tự tăng dần của trọng số và chọn $n-1$ cạnh đầu tiên.

- Nếu tập cạnh này **không tạo thành một cây khung**, thì đây chính là đáp án.
- Ngược lại, chúng tạo thành một cây khung T. Gọi $S =$ tổng trọng số của T.

Xét trường hợp đáp án được tạo bởi $n-2$ cạnh trong T và thêm vào một cạnh $e \notin T$.

Trường hợp thay thế một cạnh

Xét trường hợp đáp án được tạo bởi $n-2$ cạnh trong T và thêm vào một cạnh $e \notin T$. Khi thêm một cạnh (u, v) vào, ta cần loại bỏ cạnh có trọng số lớn nhất **không** nằm trên đường đi của (u, v), thực hiện tìm cạnh này bằng **HLD**.

Trường hợp thay thế nhiều hơn một cạnh

Đáp án tối thiểu là: $S - w_{n-1} - w_{n-2} + w_n + w_{n+1}$.
Nhận thấy nếu loại bỏ hai cạnh e_{n-2}, e_{n-1} và thêm hai cạnh e_n, e_{n+1} mà vẫn tạo thành cây khung, thì ta tồn tại cách để chỉ thay một cạnh mà tập cạnh mới không tạo thành cây (quay về trường hợp thay thế một cạnh) và tạo được một đáp án chắc chắn tốt hơn.
Do vậy, nếu ta cần thay nhiều hơn một cạnh thì đáp án là: $S - w_{n-1} - w_{n-2} + w_n + w_{n+1}$. (luôn đạt được)

Problem 4: Remove at the Lowest Cost

`https://codeforces.com/contest/2176/problem/E`

Tóm tắt đề bài

Bạn được cho một dãy gồm n phần tử, mỗi phần tử có:

- giá trị a_i,
- chi phí xoá c_i.

Mục tiêu là xoá $n-1$ phần tử sao cho tổng chi phí phải trả là nhỏ nhất, chỉ để lại đúng một phần tử. Mỗi lần thao tác, bạn chọn hai phần tử kề nhau và xoá phần tử có *giá trị nhỏ hơn*. Chi phí phải trả cho thao tác đó là *chi phí xoá nhỏ hơn* trong hai phần tử được chọn. Nếu hai phần tử có giá trị bằng nhau, bạn có thể xoá bất kỳ phần tử nào, với chi phí là chi phí nhỏ hơn của chúng. Ngoài ra, bạn có n truy vấn *zeroing*: sau truy vấn thứ i, chi phí xoá của phần tử có chỉ số p_i trở thành 0 và giữ nguyên như vậy cho các truy vấn sau.

Nhiệm vụ của bạn là:

- tính chi phí nhỏ nhất để xoá $n-1$ phần tử với chi phí ban đầu,
- và sau mỗi truy vấn zeroing.

Cần in ra $n+1$ kết quả cho mỗi test.

Ý tưởng lời giải

Bước 1: Định nghĩa hàm quy hoạch Định nghĩa $f(l, r, x)$ là chi phí nhỏ nhất để xoá toàn bộ các phần tử $a_l, a_{l+1}, \ldots, a_{r-1}$, với giả thiết rằng mỗi phần tử trong đoạn có thể bị xoá bởi một phần tử bên ngoài có chi phí xoá là x (chọn x tối ưu). Đáp án của bài toán là $f(0, n, \infty) - \infty$.

Bước 2: Phân tích các phần tử lớn nhất Gọi $p_1, p_2, \ldots, p_m$ là các vị trí có giá trị lớn nhất trong đoạn $[l, r)$. Ta có các nhận xét quan trọng sau:

- Nếu tồn tại k sao cho $l \le i < p_k < j < r$ thì hai phần tử i, j không thể xoá lẫn nhau.

- Ít nhất một trong các phần tử p_i phải bị xoá bởi một phần tử nằm ngoài đoạn.
- Các phần tử $p_1, \ldots, p_m$ chỉ có thể bị xoá với chi phí x hoặc bởi một phần tử cực đại khác.

Đặt

$$x' = \min\big(x, c_{p_1}, c_{p_2}, \ldots, c_{p_m}\big).$$

Khi đó, với lựa chọn x tối ưu, ta có công thức:

$$f(l, r, x) = x' \cdot m + f(l, p_1, x') + f(p_1 + 1, p_2, x') + \cdots + f(p_m + 1, r, x').$$

Bước 3: Tính đúng đắn Công thức trên đúng vì:

- Luôn tồn tại một phần tử cực đại bị xoá từ bên ngoài đoạn.
- Các cực đại còn lại chỉ có thể bị xoá với chi phí x'.
- Sau khi loại bỏ các cực đại, các đoạn còn lại là độc lập.

Một quy trình xoá đệ quy tương ứng có thể được xây dựng, đảm bảo luôn đạt được chi phí đúng bằng giá trị của hàm f.

Bước 4: Xử lý các truy vấn Nhận xét rằng cấu trúc các lời gọi của hàm $f(l, r, x)$ tạo thành một cây. Mỗi lần gọi f tương ứng với một đoạn $[l, r)$, và các lời gọi con tương ứng với các đoạn con sinh ra sau khi loại bỏ các phần tử có giá trị lớn nhất. Ta lưu lại cây này, đồng thời với mỗi phần tử i, ta ghi nhớ chi phí thực tế đã dùng để xoá nó trong quá trình xây dựng lời giải ban đầu. Gọi giá trị này là r_i.

Khi có một truy vấn đặt $c_{p_i} = 0$, ta thực hiện như sau:

- Xác định nút trong cây tương ứng với đoạn mà tại đó p_i là một trong các phần tử cực đại.
- Từ nút này, thực hiện duyệt theo chiều sâu (DFS) trên toàn bộ cây con của nó.
- Trong quá trình DFS, ta bỏ qua các nút (tức là các phần tử) đã được xử lý bởi các truy vấn trước đó.
- Với mỗi phần tử chưa được xử lý trong cây con, ta cập nhật giá trị $r_i = 0$, vì từ thời điểm này trở đi, phần tử đó có thể bị xoá với chi phí bằng 0.

Sau khi hoàn tất việc cập nhật, ta chỉ cần tính lại tổng các giá trị r_i để thu được đáp án mới cho truy vấn hiện tại.

Lưu ý rằng mỗi phần tử chỉ bị đưa về 0 đúng một lần trong suốt quá trình xử lý các truy vấn, do đó tổng chi phí của tất cả các lần DFS là tuyến tính theo số phần tử. Để xây dựng cây lời gọi của hàm f, ta cần thực hiện các truy vấn tìm phần tử lớn nhất trên đoạn, việc này có thể được cài đặt bằng cây đoạn (segment tree) hoặc sparse table.

Problem 5: Omega Numbers

https://codeforces.com/contest/2175/problem/F

Tóm tắt đề bài

Cho hàm $\omega(n)$ là số lượng **ước nguyên tố phân biệt** trong phân tích thừa số nguyên tố của n. Cho mảng số tự nhiên a có độ dài n và một số tự nhiên k. Định nghĩa:

$$f(a,k) = \sum_{1 \le i < j \le n} \omega(a_i \cdot a_j)^k.$$

Nhiệm vụ là tính giá trị $f(a,k)$ theo modulo 998244353.

Ý tưởng lời giải

Gọi K là số lượng ước nguyên tố phân biệt lớn nhất trong phân tích thừa số nguyên tố của một số dưới các ràng buộc đề bài. Trong bài toán này ta có:

$$K \le 6, \qquad \text{và tiệm cận } K = O(\log(\max A)).$$

Bước 1: Quy hoạch động theo GCD Ta xây dựng mảng quy hoạch động

$$dp[g][\text{sumlen}],$$

là số cặp có thứ tự (i,j) sao cho:

- $\gcd(a_i, a_j) = g$,
- $\omega(a_i) + \omega(a_j) = \text{sumlen}$.

Nếu tính được toàn bộ $dp[g][\text{sumlen}]$, thì đáp án bài toán sẽ là:

$$\sum_{g=1}^{n} \sum_{\text{sumlen}=1}^{2K} dp[g][\text{sumlen}] \cdot (\text{sumlen} - \omega(g))^k.$$

Bước 2: Đếm số phần tử chia hết cho một số Ta sử dụng một kỹ thuật quen thuộc để đếm các cặp có gcd cố định, nhưng được mở rộng cho bài toán này. Định nghĩa:

$$\text{cnt}[x][\ell]$$

là số lượng phần tử trong mảng ban đầu:

- chia hết cho x,
- và có $\omega(\text{giá trị}) = \ell$.

Để tính cnt, ta duyệt:

- mọi x từ 1 đến $\max A$,
- với mỗi x, duyệt các bội $i \cdot x$,
- cộng số lần xuất hiện của $i \cdot x$ vào $\text{cnt}[x][\omega(i \cdot x)]$.

Độ phức tạp của bước này là:

$$O(n \log n),$$

do tổng các bội tạo thành chuỗi điều hoà.

Bước 3: Tính bảng $dp[g][\textbf{sumlen}]$ Ta duyệt giá trị $\gcd = g$ từ $\max A$ giảm dần về 1. Với mỗi g, ta xét các cặp (a, b) sao cho:

$$\omega(a) = i, \quad \omega(b) = j, \quad 1 \le i, j \le K.$$

Khi đó:

$$dp[g][i + j] \mathrel{+}= \text{cnt}[g][i] \cdot \text{cnt}[g][j].$$

Cần đặc biệt chú ý trường hợp $i = j$, khi đó ta chỉ được chọn hai phần tử khác nhau:

$$dp[g][2i] \mathrel{+}= \frac{\text{cnt}[g][i] \cdot (\text{cnt}[g][i] - 1)}{2}.$$

Bước 4: Loại trừ các bội Để đảm bảo rằng $\gcd(a_i, a_j)$ đúng bằng g, ta loại trừ các cặp đã được tính cho các bội của g:

$$dp[g][\text{sumlen}] \mathrel{-}= dp[k \cdot g][\text{sumlen}], \quad \forall k \ge 2.$$

Đây chính là nguyên lý bao hàm – loại trừ trên gcd.