ĐẠI HỌC QUỐC GIA	**KỲ THI TUYỂN SINH LỚP 10 NĂM 2026**
THÀNH PHỐ HỒ CHÍ MINH	**TRƯỜNG PHỔ THÔNG NĂNG KHIẾU**
ĐÁP ÁN CHÍNH THỨC	**Môn thi: TIN HỌC**
	Ngày thi: 24/05/2026
(Đáp án gồm 03 trang)	*Thời gian làm bài: 150 phút (không kể thời gian phát đề)*

Câu 1. *(2.5 điểm)* **XÁO TRỘN XÂU**

Subtask 1: Kết quả chỉ có thể là 1 hoặc 2, do đó chỉ cần kiểm tra có thể xáo trộn xâu s để thu được xâu đối xứng hay không, tức là s có không quá một ký tự xuất hiện lẻ lần.

Subtask 2: Xét hết tất cả các cách xáo trộn xâu s. Với mỗi cách xáo trộn, tìm cách cắt nó thành ít xâu đối xứng nhất, có thể làm trong $O(|s|^2)$.

Subtask 3: Nếu tất cả các ký tự trong s đều xuất hiện chẵn lần, kết quả là 1. Ngược lại, kết quả chính là số loại ký tự xuất hiện lẻ lần trong s.

Độ phức tạp: $O(|s|)$.

Câu 2. *(2.5 điểm)* **CẢM BIẾN**

Subtask 1: Xét hết các đoạn con, với mỗi đoạn ta tính trung bình nhiệt độ và so sánh.

Subtask 2: Làm tương tự subtask 1 nhưng sử dụng mảng prefix sum để tính nhanh trung bình nhiệt độ của một đoạn.

Subtask 3: Nhiệt độ trung bình theo lý thuyết của một đoạn là:

$$l + r = 2\frac{l + (l + 1) + \cdots + r}{r - l + 1}$$

Do đó nếu đặt $b_i = a_i - 2i$ thì bài toán đưa về đếm số đoạn b có tổng bằng 0.

Sử dụng mảng prefix sum đối với mảng b, bài toán trở thành đếm số cặp phần tử bằng nhau trong mảng, có thể giải quyết bằng map hoặc sắp xếp và đếm từng đoạn.

Độ phức tạp: $O(n \log n)$.

Câu 3. *(2.5 điểm)* **SỐ RẤT LẺ**

Subtask 1. Với $n \leq 10^3$, ta có thể làm trực tiếp theo định nghĩa. Trước hết, ta có nhận xét quan trọng sau: khi xếp các ước của m lên vòng tròn, các ước chẵn và các ước lẻ phải xuất hiện xen kẽ nhau. Vì là vòng tròn, điều này xảy ra khi và chỉ khi số lượng ước chẵn bằng số lượng ước lẻ. Đồng thời ta có lẻ ước lẻ để tổng tất cả các ước là lẻ. Ta duyệt mọi ước m của n. Với mỗi m, ta lại duyệt tất cả các ước của m và tính:

- Số lượng ước lẻ;
- Số lượng ước chẵn;

- Tổng tất cả các ước.

Từ đó có thể kiểm tra được n có bao nhiêu ước là số rất lẻ. Giả sử ta thực hiện vét cạn qua mọi ước thì độ phức tạp là $O(qn^2)$.

Subtask 2. Với $n \leq 10^6$, ta vẫn dùng ý tưởng trên nhưng kết hợp thêm sàng để duyệt các ước trong $O(\log n)$. Với mỗi ước, ta lại thực hiện duyệt tiếp các ước của nó để kiểm tra điều kiện. Ta cũng có thể tiền xử lý kiểm tra trước các số rất lẻ không quá 10^6 để trả lời cho các truy vấn.

Độ phức tạp sẽ là $O(qn\log n)$.

Subtask 3. Với $n \leq 10^{12}$, ta chứng minh rằng m rất lẻ khi và chỉ khi $m = 2x^2$ với x là số nguyên dương lẻ nào đó.

Thật vậy, ta có n rất lẻ thì n phải là số chẵn (vì nó có ước chẵn). Đặt $n = 2^k u$ với u lẻ và k nguyên dương. Rõ ràng mỗi ước lẻ d của u sẽ sinh ra các ước chẵn của n là $2\mathrm{u}, 2^2\mathrm{u}, \ldots, 2^k u$. Vì thế, số lượng ước chẵn của n sẽ gấp k lần số lượng ước lẻ, kéo theo $k = 1$. Tiếp theo, ta cần u có lẻ ước. Ta viết $u = p_1^{a_1} p_2^{a_2} \ldots p_t^{a_t}$ trong đó $p_1, p_2, \ldots, p_t$ là các số nguyên tố lẻ và $a_1, a_2, \ldots, a_t$ đều nguyên dương. Khi đó số ước của u là $(a_1 + 1)(a_2 + 1) \ldots (a_t + 1)$ lẻ nên các số mũ phải đều chẵn, kéo theo u là số chính phương. Do đó bài toán trở thành: với mỗi số n, cần đếm số ước m của n có dạng $m = 2x^2$ với x lẻ.

Tiếp theo ta lại giả sử n có dạng $n = 2^e p_1^{a_1} p_2^{a_2} \ldots p_t^{a_t}$ trong đó số mũ $e \geq 0, a_1, a_2, \ldots, a_t > 0$ và $p_1, p_2, \ldots, p_t$ là các số nguyên tố lẻ.

Nếu $e = 0$ tức là n lẻ, thì không tồn tại ước rất lẻ nào của n. Nếu $e \geq 1$, ta luôn có thể chọn số mũ của 2 trong m bằng 1. Với mỗi nguyên tố lẻ p_i, số mũ của p_i trong m có thể chọn trong các số tự nhiên chẵn không vượt quá a_i. Ta có tất cả $\left\lfloor \frac{\alpha_i}{2} \right\rfloor + 1$ cách chọn số mũ của p_i. Các lựa chọn ứng với các nguyên tố lẻ là độc lập nhau nên nếu n chẵn thì đáp án là tích của các giá trị nêu trên.

Độ phức tạp: $O\left(q\sqrt{n}\right)$ vì phân tích n ra thừa số nguyên tố với chi phí $O\left(\sqrt{n}\right)$.

Subtask 4. Với giới hạn đầy đủ $q \leq 10; n \leq 10^{18}$, ta xét hết các số không vượt quá $\sqrt[3]{n}$ rồi thực hiện phân tích. Bây giờ nếu sau khi phân tích xong ta chuyển $n \to n'$ thì có các khả năng xảy ra: $n' = p$ hoặc $n' = pq$ hoặc $n' = p^2$. Ta kiểm tra số n' có phải là số chính phương hay không trong $O(1)$ và nếu có thì kết quả sẽ tăng thêm gấp đôi, các trường hợp khác không làm thay đổi kết quả.

Độ phức tạp là $O\left(q\sqrt[3]{n}\right)$.

Câu 4. *(2.5 điểm)* **THÁM HIỂM**

Subtask 1: $\mathrm{k} = 1$, $\mathrm{d}_1 = 1$. Khi $d_1 = 1$, mỗi bước đi chỉ có thể là một trong bốn hướng:

$(1,0)$, $(-1,0)$, $(0,1)$, $(0,-1)$.

Vì không có ô bị cấm và không có sói, số bước ít nhất từ $s \rightarrow t$ chính là khoảng cách Manhattan: $|x_s - x_t| + |y_s - y_t|$. Độ phức tạp: O(1).

Subtask 2: $k = 1,\ d_1 = 5$. Ở đây có các ô bị cấm và các bước đi có thể theo các hướng

$d_1 = 5 \Rightarrow (d_x, d_y) \in \{(\pm1, \pm2), (\pm2, \pm1)\}$.

Mỗi vị trí trên bản đồ được xem là một đỉnh của đồ thị có tập đỉnh V. Bài toán trở thành tìm đường đi ngắn nhất trên đồ thị không trọng số. Để xử lý vấn đề ô bị cấm, ta dùng mảng đánh dấu để kiểm tra ô có bị cấm hay không trong thời gian $O(1)$. Ta sử dụng thuật toán tìm kiếm theo chiều rộng (BFS) để xác định độ dài đường đi ngắn nhất từ điểm xuất phát s. Khi đang ở điểm (x, y), ta thử tất cả các khả năng (d_x, d_y) có dạng $n_x = x + d_x, n_y = y + d_y$. Nếu (n_x, n_y) nằm trong bản đồ, chưa được thăm và có trạng thái không bị cấm, ta đưa (n_x, n_y) vào hàng đợi. Nếu đến được điểm đích thì in ra khoảng cách nhỏ nhất, ngược lại in ra -1. BFS xét tối đa $|V|$ điểm, mỗi điểm thử 8 cách đi nên tập cạnh không quá $8|V|$.

Subtask 3: Ở đây ta có nhiều ô bị cấm và nhiều loại bước khác nhau. Từ một điểm (x, y), ta có thể đi đến điểm (u, v) nếu (u, v) nằm trong bản đồ, không bị cấm và tồn tại một giá trị d_i sao cho

$(u - x)^2 + (v - y)^2 = d_i$.

Vì $d_i \leq 26$ nên nếu đặt $d_x = u - x, d_y = v - y$, thì ta có $|d_x| \leq 5, |d_y| \leq 5$. Do đó, ta có thể sinh trước tất cả các vector bước (d_x, d_y) bằng cách duyệt $-5 \leq d_x, d_y \leq 5$ và kiểm tra xem $d_x^2 + d_y^2$ có thuộc danh sách các giá trị d_i được phép hay không. Đến đây, ta vẫn dùng BFS như trên và kiểm tra thêm điểm mới có bị cấm hay không vẫn bằng mảng đánh dấu. Gọi S là tập các cách dịch chuyển thì bằng cách sinh trực tiếp, ta thấy kích thước tập này không quá lớn. Độ phức tạp: $O(|V||S|)$.

Subtask 4: Ở subtask này ta chỉ có một con sói, ta có thể coi nó như ô bị cấm. Ở mỗi lượt chơi, các ô bị cấm tạo ra bởi con sói này sẽ loang ra trên bảng qua các ô kề cạnh. Như vậy, ta sẽ tiền xử lý, thực hiện BFS từ ô có sói ra toàn bảng để xem với mỗi ô thì thời điểm đầu tiên mà sói có thể tiếp cận là bao nhiêu, đặt mảng đó là $time[x][y]$. Mỗi ô của sói cũng tiếp cận 4 vị trí xung quanh và mỗi ô của bảng được duyệt qua không quá 1 lần nên chi phí loang vẫn là $O(mn)$. Như vậy, ở lượt đi thứ t của người chơi, một ô vuông có thể tiếp cận từ sói sau không quá t lượt cũng trở thành ô bị cấm. Cụ thể hơn, ta điều chỉnh cách BFS cũ, ta chỉ đưa các ô (x, y) vào hàng đợi ở thời điểm thứ t nếu như $time[x][y] < t$.

Độ phức tạp: $O(mn + |V||S|)$.

Subtask 5: Khi có nhiều sói và có các ô bị cấm, ta vẫn thực hiện tương tự trên. Ở bước đầu của BFS, ta đưa danh sách các ô có sói vào hàng đợi. Ở mỗi bước sau, ta vẫn loang ra từ danh sách trước đó và có kiểm tra bỏ qua các ô bị cấm. Như vậy ở đây, $time[x][y]$ sẽ chính là thời gian ít nhất để một con sói nào đó tiếp cận được ô (x, y). Đoạn sau thao tác cho người chơi thực hiện như trên.

Độ phức tạp: $O(mn + |V||S|)$.