

## Tóm tắt bài toán

Cho  $N$  cột sỏi, cột thứ  $i$  có  $s_i$  viên. Mỗi nước đi, người chơi chọn một cột có  $s_i > 0$  và được bốc tối đa:

$$k \leq \lfloor \log_2(s_i) \rfloor + 1.$$

Không đi được sẽ thua, Đạt đi trước.

Bài toán là một *Impartial Combinatorial Game*. Vì vậy, kết quả phụ thuộc vào *giá trị Grundy* của từng cột:

$$\text{Đạt thắng} \iff \bigoplus_{i=1}^N G(s_i) \neq 0.$$

---

### Subtask 1: $s_i \leq 10^5$

#### Ý tưởng

Với giới hạn nhỏ, có thể tính giá trị Grundy cho mọi  $x$  từ 1 đến  $10^5$  thông qua công thức trực tiếp:

$$G(x) = \text{mex}\{G(x - k) \mid 1 \leq k \leq \lfloor \log_2(x) \rfloor + 1\}.$$

#### Phân tích độ phức tạp

Với mỗi  $x$ , số bước hợp lệ là  $\Theta(\log x)$ , nên độ phức tạp thời gian của bài toán là:

$$T(n) = \sum_{x=1}^{\max s_i} \Theta(\log x) \approx \Theta(\max s_i \log(\max s_i))$$

Vì cần tạo mảng Grundy gồm  $10^5$  phần tử, nên độ phức tạp không gian của bài toán là:

$$\Theta(\max s_i).$$

---

### Subtask 2: $s_i \leq 10^9$

Để tránh phải tính toàn bộ Grundy đến  $10^9$ , thuật toán sử dụng tính chất sau.

Cho:

$$\text{base} = 2^{\text{bit\_length}(x)-1}, \quad m = \text{bit\_length}(x) + 1,$$

$$\text{target} = \text{base} + (x - \text{base}) \bmod m.$$

Nếu  $x > \text{target}$  thì ta có rút gọn:

$$G(x) = G(\text{target}).$$

Ngược lại:

$$G(x) = \text{mex}\{G(x - k) \mid 1 \leq k \leq \text{bit\_length}(x)\}.$$

## Chứng minh tính đúng đắn thuật toán

Đặt  $b = \text{bit\_length}(x)$ , khi đó mọi số có  $b$  bit đều thuộc đoạn:

$$I_b = [2^{b-1}, 2^b - 1].$$

Trong đoạn này, mọi trạng thái đều có đúng  $b$  nước đi hợp lệ: có thể trừ lần lượt 1, 2, …,  $b$ . Điều này dẫn đến cấu trúc chuyển tiếp của mọi trạng thái trong cùng đoạn  $I_b$  gần như giống hệt nhau, chỉ dịch sang trái đúng 1 đơn vị khi tăng giá trị  $x$ . Do tính chất này, giá trị Grundy trong nhóm phải tuân hoà với chu kỳ đúng bằng số lượng trạng thái mà một trạng thái có thể sinh ra, tức là chu kỳ  $b + 1$ .

Ta xét biểu diễn  $x$  theo khoảng và phần dư:

$$x = \text{base} + t, \quad \text{base} = 2^{b-1}, \quad 0 \leq t < 2^{b-1}.$$

Do Grundy lặp lại theo chu kỳ  $b + 1$ , nên hai trạng thái  $x$  và

$$x' = \text{base} + (t \bmod (b + 1))$$

thuộc cùng vị trí trong chu kỳ. Vì các tập trạng thái có thể đi đến từ  $x$  và  $x'$  chỉ khác nhau bởi dịch chuyển đồng nhất trong đoạn  $I_b$ , nên tập các giá trị Grundy suy ra (tức là tập  $\{G(x - k)\}$ ) giống nhau về cấu trúc. Toán tử mex chỉ phụ thuộc vào cấu trúc này, không phụ thuộc vào giá trị tuyệt đối của  $x$ . Do đó:

$$G(x) = G(x').$$

Nếu  $x > x'$  thì thay  $x$  bằng  $x'$  là hợp lệ vì đưa trạng thái về “đại diện chuẩn” trong chu kỳ của nó.

## Độ phức tạp thời gian và không gian

Vì sử dụng `lru_cache`, mỗi trạng thái Grundy chỉ được tính đúng một lần. Do mọi  $x$  sau bước rút gọn đều rơi vào một tập rất nhỏ có kích thước tối đa  $b + 1 \leq \log_2(\max s_i) + 1$ , tổng số trạng thái Grundy khác nhau mà hàm `compute_nim_value` có thể sinh ra là

$$M = \sum_{b=1}^{\log_2(\max s_i)} (b + 1) = \Theta((\log \max s_i)^2).$$

Khi tính một trạng thái Grundy mới, ta có hai bước:

- Bước rút gọn

$$x \rightarrow x' = \text{base} + (x - \text{base}) \bmod (b + 1)$$

được thực hiện trong thời gian  $O(1)$ .

- Nếu không rút gọn được, ta tính trực tiếp

$$G(x) = \text{mex}\{G(x - k) \mid 1 \leq k \leq b\},$$

với  $b = \text{bit\_length}(x) \leq \log_2(\max s_i)$ . Do đó, mỗi trạng thái cần thời gian

$$O(b) = O(\log \max s_i).$$

Suy ra tổng thời gian để xây dựng toàn bộ bảng giá trị Grundy (tất cả cache) là

$$T_{\text{cache}} = M \cdot O(\log \max s_i) = \Theta((\log \max s_i)^3).$$

Sau khi cache đã đầy đủ, việc truy vấn  $G(s_i)$  cho mỗi cột chỉ cần thời gian  $O(1)$  nhờ tra cứu trực tiếp, nên xử lý tất cả  $N$  cột đầu vào tốn

$$T_{\text{input}} = \Theta(N).$$

Do đó, tổng thời gian của thuật toán là

$$T_{\text{total}} = \Theta(N) + \Theta((\log \max s_i)^3).$$

Cuối cùng, tổng bộ nhớ cần thiết chính là số trạng thái Grundy được lưu trong cache và  $N$  phần tử cần xét, độ phức tạp không gian của bài toán là:

$$\text{Không gian} = \Theta(N + (\log \max s_i)^2) \leq \Theta(N)$$