

Codeforces 318A - Even Odds

Rèn luyện tư duy toán học (Math vs Brute Force)

Coach Tư Duy Thuật Toán

Ngày 13 tháng 1 năm 2026

Lời chào

Coach Tư Duy Thuật Toán

Chào mừng bạn. Rất tốt, bạn đã chọn bài **Codeforces 318A - Even Odds**.

Đây là một bài toán kinh điển để rèn luyện tư duy toán học (Math) thay vì dùng vòng lặp (Brute Force).

Chúng ta sẽ không vội vàng viết code. Hãy cùng nhau “mổ xẻ” nó ngay bây giờ.

Nội dung chính

Bước 1: Tiếp nhận & Phẫu thuật (Briefing)

Tên bài toán: Even Odds (Chẵn Lẻ).

Đề bài (Ngôn ngữ con người)

Người ta viết các số nguyên từ 1 đến n lên bảng. Nhưng họ không viết theo thứ tự bình thường (1, 2, 3...). Họ viết theo quy tắc "**Lẻ trước, Chẵn sau**".

- ① Đầu tiên là tất cả các số lẻ (tăng dần).
- ② Sau đó mới đến tất cả các số chẵn (tăng dần).

Nhiệm vụ: Tìm xem con số nằm ở vị trí thứ k trong dãy số mới này là số mấy?

Lộ trình tư duy

Để giải quyết bài này, chúng ta cần đi qua 2 mảng ghép logic (Micro-Chunks):

- **Chunk 1: Tìm “Biên giới”**

(Có bao nhiêu số lẻ? Khi nào thì bước sang vùng số chẵn?).

- **Chunk 2: Công thức truy xuất**

(Nếu ở vùng Lẻ thì tính sao? Nếu ở vùng Chẵn thì tính sao?).

Bạn đã sẵn sàng bước vào mảng ghép đầu tiên chưa?

1. Logic (Explain)

Hãy tưởng tượng dãy số từ 1 đến n giống như một lớp học có n học sinh. Thầy giáo chia lớp thành 2 nhóm:

- **Nhóm 1 (Nhóm Lẻ):** Gồm các bạn mang số báo danh 1, 3, 5... xếp hàng trước.
- **Nhóm 2 (Nhóm Chẵn):** Gồm các bạn mang số báo danh 2, 4, 6... xếp hàng nối đuôi phía sau.

Vấn đề cốt lõi đầu tiên: Để biết vị trí thứ k nằm ở Nhóm 1 hay Nhóm 2, ta phải biết **Nhóm 1 có tất cả bao nhiêu thành viên**.
Ta gọi số lượng thành viên nhóm lẻ là P (Partition point).

2. Bẫy (Trap) **WARN**

Sai lầm thường gặp

Rất nhiều bạn mặc định chia đôi: $P = n/2$.

- Với $n = 10$, số lẻ là $\{1, 3, 5, 7, 9\} \rightarrow$ có 5 số. ($10/2 = 5 \rightarrow$ Đúng).
- Nhưng với $n = 7$, số lẻ là $\{1, 3, 5, 7\} \rightarrow$ có 4 số.
- Nếu dùng phép chia số nguyên trong máy tính: $7/2 = 3 \rightarrow$ **SAI** (Thiếu mất 1 người).

3. Thủ thách (Challenge) - Phần 1

Câu hỏi

Giúp tôi xác định số lượng phần tử lẻ (P) trong các trường hợp sau:

- ① Nếu $n = 4$ (Dãy: 1, 3, 2, 4).
- ② Nếu $n = 5$ (Dãy: 1, 3, 5, 2, 4).

Câu hỏi chốt: Công thức toán học nào đúng để tính số lượng số lẻ P cho mọi n ?

- A. $n/2$
- B. $(n - 1)/2$
- C. $(n + 1)/2$ (Lấy phần nguyên)

3. Thủ thách (Challenge) - Đáp án

Đáp án:

3. Thủ thách (Challenge) - Đáp án

Đáp án:

Chính xác! **Đáp án C** là chuẩn nhất.

Trong lập trình C++ (và nhiều ngôn ngữ khác), phép chia số nguyên (`int / int`) sẽ tự động làm tròn xuống (ví dụ $3.5 \rightarrow 3$).

Vì vậy, công thức:

$$P = (n + 1)/2$$

chính là mẹo toán học để “làm tròn lên”, đảm bảo tính đúng số lượng cho cả trường hợp n lẻ và n chẵn.

*Tuyệt vời. Bạn đã nắm được biến quan trọng nhất: **Biên giới P**.*

1. Logic (Explain)

Bây giờ chúng ta cầm trong tay tấm vé số thứ tự k . Chúng ta cần biết mình phải đi vào **Cửa số 1 (Nhóm Lẻ)** hay **Cửa số 2 (Nhóm Chẵn)**.
Quy trình: So sánh vị trí k với biên giới $P = (n + 1)/2$.

Trường hợp 1: Nhóm Lẻ ($k \leq P$)

Nếu $k \leq P$ (Vị trí nằm trong nửa đầu):

- Dãy số là: 1, 3, 5, 7...
- Vị trí thứ 1 → giá trị 1 ($2 \times 1 - 1$)
- Vị trí thứ 2 → giá trị 3 ($2 \times 2 - 1$)

Công thức tổng quát

$$\text{Value} = 2 \times k - 1$$

Trường hợp 2: Nhóm Chẵn ($k > P$)

Nếu $k > P$ (Vị trí nằm ở nửa sau):

- Lúc này, ta đã bước qua hết P số lẻ.
- Vị trí **thực sự** của ta trong nhóm chẵn là: $new_k = k - P$.
- Dãy số chẵn: 2, 4, 6, 8...
- Vị trí thứ 1 trong nhóm chẵn \rightarrow giá trị 2 (2×1)

Công thức tổng quát

$$\text{Value} = 2 \times (k - P)$$

2. Bẫy (Trap) **TLE**

Sai lầm chết người

Ở bài này, giới hạn của n và k lên tới 10^{12} (1000 tỷ).

- Dùng vòng lặp for (`int i = 1...`) để đếm.
- **Hậu quả:** Máy tính chạy quá 1 giây → **Time Limit Exceeded (TLE)**.

Giải pháp: Bắt buộc dùng công thức $O(1)$ như trên, không được dùng vòng lặp. Và kiểu dữ liệu phải là long long.

3. Thủ thách (Challenge) - Áp dụng

Hãy áp dụng 2 công thức trên để giải quyết tình huống sau.

Dữ liệu: Cho $n = 10, k = ?$. (Ta đã biết $P = 5$).

Câu hỏi

- ① Nếu $k = 3$: Kết quả là bao nhiêu?
- ② Nếu $k = 7$: Kết quả là bao nhiêu? (Gợi ý: $7 > 5$, hãy tìm vị trí tương đối trước).

3. Thủ thách (Challenge) - Áp dụng

Hãy áp dụng 2 công thức trên để giải quyết tình huống sau.

Dữ liệu: Cho $n = 10, k = ?$. (Ta đã biết $P = 5$).

Câu hỏi

- ① Nếu $k = 3$: Kết quả là bao nhiêu?
- ② Nếu $k = 7$: Kết quả là bao nhiêu? (Gợi ý: $7 > 5$, hãy tìm vị trí tương đối trước).

Kết quả:

- **Logic 1:** $k = 3 \leq 5$ (Vùng Lẻ) $\rightarrow 2 \times 3 - 1 = 5$.
- **Logic 2:** $k = 7 > 5$ (Vùng Chẵn) \rightarrow Bỏ qua 5 số lẻ, tìm số chẵn thứ $(7 - 5) = 2$. $\rightarrow 2 \times 2 = 4$.

Hoàn hảo!

Chunk 3: Bẫy dữ liệu (Data Type Trap)

Đây là nơi 50% người mới (Newbies) bị “ăn hành” (Wrong Answer hoặc Runtime Error) dù thuật toán đúng.

Dữ kiện:

- Đề bài cho n và k có thể lên tới 10^{12} (1000 tỷ).
- Kiểu int trong C++ chỉ chứa được tối đa khoảng 2×10^9 (2 tỷ).

Kết luận

Nếu bạn khai báo `int n, k;`, biến của bạn sẽ bị tràn số (Overflow) giống như đổ 10 lít nước vào cái chai 2 lít vậy. Kết quả sẽ ra sai lệch hoàn toàn.

Giải pháp: Bắt buộc dùng `long long` (chứa được tới 9×10^{18}).

Tổng kết thuật toán (The Blueprint)

Đây là bản thiết kế để bạn chuyển ngữ sang C++:

① **Input:** Nhập n, k (Nhớ dùng long long).

② **Bước 1 (Tính P):**

```
long long part = (n + 1) / 2;
```

③ **Bước 2 (Kiểm tra vị trí):**

- **Nếu** $k \leq part$: In ra $2 \times k - 1$.

- **Ngược lại (Else):** In ra $2 \times (k - part)$.

④ **Output:** Kết quả.

Bạn đã có đủ mọi “nguyên liệu”!

- ① Công thức tính biên giới.
- ② Công thức tính giá trị từng vùng.
- ③ Loại dữ liệu cần dùng.

Bạn có muốn tự tay viết đoạn code này để tôi kiểm tra (Review), hay bạn muốn tôi cung cấp Code mẫu chuẩn (Best Practice) ngay bây giờ?

Cảm ơn bạn đã theo dõi!