

Phân tích Tư duy Thuật toán: Codeforces 630A Again Twenty Five! - LHTL Edition

Coach Tư Duy Thuật Toán

Ngày 13 tháng 1 năm 2026

Giới thiệu

Chúng ta sẽ khởi động ngay với bài toán **Codeforces 630A - Again Twenty Five!**

Đây là một bài toán cực kỳ thú vị vì nó là ví dụ điển hình cho việc: "*Code trâu bò (Brute force) thì chết, mà tư duy đúng thì chỉ mất 1 giây*".

Bước 1: Tiếp nhận & Phẫu thuật đề bài (Briefing)

1. Tóm tắt đề bài (Ngôn ngữ con người)

Đề bài cho bạn một số nguyên n (với $n \geq 2$). Số n này có thể cực kỳ lớn (lên tới $2 \cdot 10^{18}$).

Nhiệm vụ của bạn là tính kết quả của phép tính lũy thừa 5^n , nhưng **chỉ cần in ra 2 chữ số tận cùng** của kết quả đó.

2. Lộ trình tư duy (Roadmap)

Để giải quyết bài này, chúng ta cần đi qua 2 mảnh ghép tư duy:

- **Chunk 1:** Quan sát quy luật "cái đuôi" của số 5.
- **Chunk 2:** Đối mặt với cạm bẫy "Số khổng lồ" (Big Integer).

Chunk 1: Truy tìm quy luật (Logic & Bẫy)

Logic (Ân dụ)

Hãy tưởng tượng bạn có một cái máy tính cũ kỹ bị hỏng màn hình. Dù bạn tính ra con số hàng tỷ tỷ, màn hình của nó bé xíu chỉ hiển thị được **đúng 2 con số cuối cùng** bên phải mà thôi.

Thay vì cố gắng tính toán những con số khổng lồ, chúng ta hãy thử làm "thám tử" bằng cách tính tay vài phép tính nhỏ.

Bẫy (Trap)

Sai lầm chết người là nghĩ: "*Phải dùng vòng lặp nhân n lần*" hoặc "*Dùng hàm 'pow(5, n)'*".

Nếu n là 1 tỷ tỷ, máy tính sẽ chạy đến hết đời hoặc bị tràn bộ nhớ (Overflow).

Chunk 1: Thủ thách tư duy (Mental Check)

Hãy lấy giấy bút và điền vào chỗ trống (chỉ quan tâm 2 số cuối):

- ① $5^2 (5 \times 5) = 25 \rightarrow$ 2 số cuối là: **25**
- ② $5^3 (25 \times 5) = 125 \rightarrow$ 2 số cuối là: **??**
- ③ $5^4 (125 \times 5) = 625 \rightarrow$ 2 số cuối là: **??**

Chunk 1: Thủ thách tư duy (Mental Check)

Hãy lấy giấy bút và điền vào chỗ trống (chỉ quan tâm 2 số cuối):

- ① $5^2 (5 \times 5) = 25 \rightarrow$ 2 số cuối là: **25**
- ② $5^3 (25 \times 5) = 125 \rightarrow$ 2 số cuối là: **??**
- ③ $5^4 (125 \times 5) = 625 \rightarrow$ 2 số cuối là: **??**

Kết quả phân tích

Chính xác tuyệt đối! Bạn đã phát hiện ra "lời nguyền" của con số 25. Cứ đuôi là 25 mà đem nhân với 5, thì kết quả mới lại sinh ra đuôi 125 (tức là 2 số cuối vẫn hoàn lại là 25). \rightarrow Nó là một vòng lặp vĩnh cửu.

Chunk 2: Đổi mặt với Input (Ignore The Beast)

Logic (Ấn dụ)

Hãy tưởng tượng n là tiếng gầm của một con quái vật. Con quái vật có thể gầm to (n cực lớn) hoặc gầm nhỏ (n nhỏ), nhưng **phản ứng** của bạn chỉ có một chiêu duy nhất là ném ra lá bùa số "25".

n chỉ đóng vai trò là "tín hiệu bắt đầu". Giá trị cụ thể của nó **vô nghĩa**.

Bẫy (Trap)

Rất nhiều bạn lo lắng: "*Dùng int có tràn không?*", "*Có cần dùng string không?*"

Sự thật là: Nếu bạn không dùng n để tính toán, bạn không cần quan tâm nó lớn thế nào!

Chunk 2: Thủ thách tư duy (Mental Check)

Giả sử Input: 999999999999999999. Chọn phương án xử lý:

- **A:** Dùng string đọc vào, kiểm tra độ dài, in 25.
- **B:** Khai báo biến tạm (long long n), đọc input cho đúng thủ tục, sau đó **kết xác nó** và in ra 25.
- **C:** Không thèm đọc input, in luôn 25.

Chunk 2: Thủ thách tư duy (Mental Check)

Giả sử Input: 9999999999999999. Chọn phương án xử lý:

- **A:** Dùng string đọc vào, kiểm tra độ dài, in 25.
- **B:** Khai báo biến tạm (long long n), đọc input cho đúng thủ tục, sau đó **kết quả** và in ra 25.
- **C:** Không thèm đọc input, in luôn 25.

Tại sao C "nguy hiểm"? (The "Sync" Trap)

- Nếu bạn chọn C: Bạn lờ đi tờ giấy Input. Nó vẫn nằm trong bộ nhớ đệm (buffer).
- Nếu đề có nhiều test cases: Bạn sẽ bị "lệch nhịp" ở các bài sau.

→ Phương án B là "Chuẩn cơm mẹ nấu".

Bước cuối: Tổng kết chiến thuật (The Solution)

Mã giả (Pseudocode)

- ① **Chuẩn bị:** Tạo một chỗ chứa (biến) để nhận quả bom n (dù biết thừa không dùng). Dùng long long hoặc string.
- ② **Nhập:** Đọc giá trị từ bàn phím vào biến đó.
- ③ **Xử lý:** Không làm gì cả! (Bước này trông).
- ④ **Xuất:** In ra màn hình số 25.

Code C++ "Chuẩn mực"(Tham khảo)

Đây là cách chúng ta hiện thực hóa tư duy trên:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // 1. Chuẩn bị biến (Dung long long cho an toàn)
6     long long n;
7
8     // 2. Đọc input (Nhận hàng cho dung thu tuc)
9     cin >> n;
10
11    // 3. In ra kết quả (Bat chap n là gì)
12    cout << 25;
13
14    return 0;
15 }
```

Lưu ý: Bạn cũng có thể dùng `string n; cin » n;` để an toàn tuyệt đối.



Chúc mừng!

Bạn đã giải quyết xong bài toán bằng tư duy thay vì sức mạnh cơ bắp.

Bạn có muốn thử sức ngay với một bài toán khác đòi hỏi tư duy logic "lắt léo" hơn một chút không? (Ví dụ: Codeforces 4A - Watermelon)?