

Codeforces 1684A - Digit Minimization

Huấn luyện viên Tư duy Thuật toán

Slide Learning C++

Ngày 21 tháng 1 năm 2026

Lời mở đầu

Triết lý

Chào bạn! Tôi đã sẵn sàng nhập vai **Huấn luyện viên Tư duy Thuật toán** theo triết lý "Learning How to Learn". Chúng ta sẽ cùng nhau "mổ xẻ" bài toán để tìm ra bản chất nhé.

Bước 1: Phân thuật đề bài (Deconstruct)

Cốt lõi của vấn đề:

- **Dữ liệu vào:** Một số nguyên dương n .
- **Người 1:** Chọn hai vị trí khác nhau trong số n và hoán đổi (swap).
- **Người 2:** Xóa chữ số cuối cùng bên phải của số đó.
- **Kết thúc:** Trò chơi dừng khi số chỉ còn **đúng một chữ số**.
- **Mục tiêu:** Tìm chữ số cuối cùng còn sót lại là **nhỏ nhất có thể**.

Lộ trình tư duy (Chunking)

- ❶ **Chunk 1:** Quan sát quy luật khi số chỉ có đúng 2 chữ số.
- ❷ **Chunk 2:** Quan sát quy luật khi số có từ 3 chữ số trở lên.
- ❸ **Chunk 3:** Tổng kết chiến thuật tối ưu.

Sẵn sàng?

Bạn đã sẵn sàng mổ xẻ mảnh ghép đầu tiên chưa?

Chunk 1: Trường hợp số có đúng 2 chữ số

Bẫy logic

Bạn **bắt buộc** phải hoán đổi trước khi bị xóa. Không có quyền lựa chọn "không hoán đổi".

Ví dụ: Số $n = 12$

- 1 Sau khi Người 1 hoán đổi, số trở thành bao nhiêu?
- 2 Sau đó Người 2 xóa chữ số cuối, kết quả cuối cùng là gì?

Chunk 1: Trường hợp số có đúng 2 chữ số

Bẫy logic

Bạn **bắt buộc** phải hoán đổi trước khi bị xóa. Không có quyền lựa chọn "không hoán đổi".

Ví dụ: Số $n = 12$

- 1 Sau khi Người 1 hoán đổi, số trở thành bao nhiêu?
- 2 Sau đó Người 2 xóa chữ số cuối, kết quả cuối cùng là gì?

Giải đáp

Kết quả là: **2**. Với số có 2 chữ số, đáp án luôn là **chữ số thứ hai** (hàng đơn vị).

Chunk 2: Khi số có từ 3 chữ số trở lên

Ví dụ: Số $n = 312$

- Người 1 có thể chọn **bất kỳ** 2 vị trí nào để hoán đổi.
- Người 2 xóa chữ số cuối cùng bên phải.

Thử thách tư duy

Nếu mục tiêu là giữ lại số **1** đến cuối cùng, bạn nên đưa số 1 về vị trí nào để nó không bao giờ bị "bay màu"?

Chunk 2: Khi số có từ 3 chữ số trở lên

Ví dụ: Số $n = 312$

- Người 1 có thể chọn **bất kỳ** 2 vị trí nào để hoán đổi.
- Người 2 xóa chữ số cuối cùng bên phải.

Thử thách tư duy

Nếu mục tiêu là giữ lại số **1** đến cuối cùng, bạn nên đưa số 1 về vị trí nào để nó không bao giờ bị "bay màu"?

Chiến thuật

Đưa chữ số nhỏ nhất về một vị trí an toàn (không phải vị trí cuối) cho đến tận bước cuối cùng.

Tổng kết chiến thuật

Trường hợp	Chiến thuật	Kết quả
Số có 2 chữ số	Bị ép buộc hoán đổi	Chữ số thứ 2
Số có 3+ chữ số	Tự do điều khiển	Chữ số nhỏ nhất

Bước 3: Tổng kết & Chốt thuật toán

Mã giả (Pseudocode)

1. nhập vào chuỗi ký tự S (đại diện cho số n)
2. nếu **do** dài của S bằng 2:
 in ra ký tự S[1] (ký tự thứ hai)
3. ngược lại (nếu **do** dài lớn hơn 2):
 tìm ký tự nhỏ nhất trong chuỗi S
 in ra ký tự nhỏ nhất **do**

Thử thách cuối cùng

Hãy thử áp dụng thuật toán cho 2 test case này:

① $n = 249$

② $n = 42$

Thử thách cuối cùng

Hãy thử áp dụng thuật toán cho 2 test case này:

- 1 $n = 249$
- 2 $n = 42$

Đáp án

- Với $n = 249$: Đáp án là **2** (nhỏ nhất).
- Với $n = 42$: Đáp án là **2** (vị trí thứ hai).

Tiếp theo

Bạn đã hoàn toàn làm chủ được tư duy bài toán này! Bạn có muốn mình giúp chuyển mã giả này sang một ngôn ngữ lập trình cụ thể (C++, Python...) hay mở xẻ bài toán khác thú vị hơn?