

Phẫu thuật bài toán: Nearest Interesting Number

Algorithmic Coach - Codeforces 1183/A

Slide Learning CPP

Ngày 21 tháng 1 năm 2026

Chào mừng bạn!

Triết lý học tập

Chúng ta sẽ cùng chinh phục thử thách trên Codeforces theo phương pháp "**Learning How to Learn**". Thay vì những dòng code khô khan, chúng ta sẽ "phẫu thuật" bài toán để tìm ra chiến thuật tối ưu.

1. Phẫu thuật đề bài (Deconstruct)

Cốt lõi vấn đề

- **Khái niệm "Số thú vị":** Một số có tổng các chữ số chia hết cho 4.
- **Nhiệm vụ:** Cho số nguyên n , tìm số nguyên a nhỏ nhất sao cho:
 - ① $a \geq n$
 - ② a là một "số thú vị".

Lộ trình tư duy

- **Chunk 1:** Cách tính tổng các chữ số.
- **Chunk 2:** Chiến thuật tìm kiếm (Dùng công thức hay tiến lên?).
- **Chunk 3:** Xử lý bẫy và chốt thuật toán.

2. Mảng ghép 1: Chiếc máy quét tổng (Digit Sum)

Ân dụ: Túi tiền

Hãy tưởng tượng số n như một túi tiền có các tờ tiền mệnh giá khác nhau. Để tính tổng, bạn phải lấy từng tờ ra và cộng lại.

Bẫy logic

Phân biệt giá trị số (ví dụ: 43) và tổng các chữ số ($4 + 3 = 7$). Vì 7 không chia hết cho 4, số 43 không "thú vị".

Thử thách tư duy

Giả sử ta có số $n = 91$:

- ➊ Tổng các chữ số của n là bao nhiêu?
- ➋ n có phải là "số thú vị" không?

2. Mảng ghép 1: Chiếc máy quét tổng (Digit Sum)

Ân dụ: Túi tiền

Hãy tưởng tượng số n như một túi tiền có các tờ tiền mệnh giá khác nhau. Để tính tổng, bạn phải lấy từng tờ ra và cộng lại.

Bẫy logic

Phân biệt giá trị số (ví dụ: 43) và tổng các chữ số ($4 + 3 = 7$). Vì 7 không chia hết cho 4, số 43 không "thú vị".

Thử thách tư duy

Giả sử ta có số $n = 91$:

- ➊ Tổng các chữ số của n là bao nhiêu?
- ➋ n có phải là "số thú vị" không?

Đáp án: Tổng là $9 + 1 = 10$. Không thú vị vì 10 không chia hết cho 4.

3. Mảng ghép 2: Chiến thuật "Bước từng bước"

Câu hỏi

Cần công thức siêu việt để "nhảy" thẳng đến đáp án hay chỉ cần đi bộ?

Ân dụ: Giao hàng trên phố

Bạn đứng ở cửa nhà số n . Hãy tìm ngôi nhà "màu xanh" (thú vị) đầu tiên bằng cách kiểm tra $n, n + 1, n + 2 \dots$

3. Mảng ghép 2: Chiến thuật "Bước từng bước"

Câu hỏi

Cần công thức siêu việt để "nhảy" thẳng đến đáp án hay chỉ cần đi bộ?

Ân dụ: Giao hàng trên phố

Bạn đứng ở cửa nhà số n . Hãy tìm ngôi nhà "màu xanh" (thú vị) đầu tiên bằng cách kiểm tra $n, n + 1, n + 2 \dots$

Thực hành với $n = 432$

- $432: 4 + 3 + 2 = 9$ (Không)
- $433: 4 + 3 + 3 = 10$ (Không)

3. Mảng ghép 2: Chiến thuật "Bước từng bước"

Câu hỏi

Cần công thức siêu việt để "nhảy" thẳng đến đáp án hay chỉ cần đi bộ?

Ân dụ: Giao hàng trên phố

Bạn đứng ở cửa nhà số n . Hãy tìm ngôi nhà "màu xanh" (thú vị) đầu tiên bằng cách kiểm tra $n, n + 1, n + 2 \dots$

Thực hành với $n = 432$

- $432: 4 + 3 + 2 = 9$ (Không)
- $433: 4 + 3 + 3 = 10$ (Không)
- $434: 4 + 3 + 4 = 11$ (Không)

3. Mảng ghép 2: Chiến thuật "Bước từng bước"

Câu hỏi

Cần công thức siêu việt để "nhảy" thẳng đến đáp án hay chỉ cần đi bộ?

Ân dụ: Giao hàng trên phố

Bạn đứng ở cửa nhà số n . Hãy tìm ngôi nhà "màu xanh" (thú vị) đầu tiên bằng cách kiểm tra $n, n + 1, n + 2 \dots$

Thực hành với $n = 432$

- $432: 4 + 3 + 2 = 9$ (Không)
- $433: 4 + 3 + 3 = 10$ (Không)
- $434: 4 + 3 + 4 = 11$ (Không)
- $435: 4 + 3 + 5 = 12$ (Đúng! 12:4)

Kết luận: Khoảng cách giữa các số thú vị rất ngắn (thường ≤ 5 bước).

4. Mảnh ghép 3: Tổng kết thuật toán (The Blueprint)

Chiến thuật: Vòng lặp "Thử và Sai"

Vì khoảng cách nhỏ, vòng lặp đơn giản là tối ưu nhất.

```
1 Ham TinhTongChuSo(x):
2     tong = 0
3     Trong khi x > 0:
4         tong = tong + (x chia lay du cho 10)
5         x = x chia lay nguyen cho 10
6     Tra ve tong
7
8 Nhap n
9 Trong khi TinhTongChuSo(n) khong chia het cho 4:
10    n = n + 1
11
12 In ra n
```

Listing 1: Mã giả tư duy

5. Thủ thách cuối cùng (Final Boss)

Trường hợp bẫy: $n = 99$

Hãy cẩn thận khi bước từ 99 sang 100!

- $99 \rightarrow 9 + 9 = 18$ (Không)

5. Thủ thách cuối cùng (Final Boss)

Trường hợp bẫy: $n = 99$

Hãy cẩn thận khi bước từ 99 sang 100!

- $99 \rightarrow 9 + 9 = 18$ (Không)
- $100 \rightarrow 1 + 0 + 0 = 1$ (Không)

5. Thủ thách cuối cùng (Final Boss)

Trường hợp bẫy: $n = 99$

Hãy cẩn thận khi bước từ 99 sang 100!

- $99 \rightarrow 9 + 9 = 18$ (Không)
- $100 \rightarrow 1 + 0 + 0 = 1$ (Không)
- $101 \rightarrow 1 + 0 + 1 = 2$ (Không)

5. Thủ thách cuối cùng (Final Boss)

Trường hợp bẫy: $n = 99$

Hãy cẩn thận khi bước từ 99 sang 100!

- $99 \rightarrow 9 + 9 = 18$ (Không)
- $100 \rightarrow 1 + 0 + 0 = 1$ (Không)
- $101 \rightarrow 1 + 0 + 1 = 2$ (Không)
- $102 \rightarrow 1 + 0 + 2 = 3$ (Không)

5. Thủ thách cuối cùng (Final Boss)

Trường hợp bẫy: $n = 99$

Hãy cẩn thận khi bước từ 99 sang 100!

- $99 \rightarrow 9 + 9 = 18$ (Không)
- $100 \rightarrow 1 + 0 + 0 = 1$ (Không)
- $101 \rightarrow 1 + 0 + 1 = 2$ (Không)
- $102 \rightarrow 1 + 0 + 2 = 3$ (Không)
- $103 \rightarrow 1 + 0 + 3 = 4$ (**Thú vị!**)

Kết luận

Dù số chữ số có thay đổi, logic "thử từng số" vẫn hoạt động hoàn hảo.

Tổng kết

- **Kỹ thuật:** Sử dụng vòng lặp 'while' kết hợp hàm tính tổng chữ số.
- **Lưu ý:** Khi viết hàm TinhTongChuSo, hãy dùng biến tạm để tránh làm mất giá trị gốc của n.

Bước tiếp theo

Bạn có muốn tôi hỗ trợ chuyển ý tưởng này sang C++ hay Python không, hay bạn muốn tự mình "múa phím" thử xem sao?