Giải bài 1411B - Fair Numbers

Slide Learning C++

Ngày 21 tháng 1 năm 2026

Tiếp nhận & Phẫu thuật bài toán

Dữ kiện cốt lõi

Tìm số nguyên x nhỏ nhất sao cho $x \geq n$ và x là một **"số công bằng"(fair number)**.

Định nghĩa Số công bằng

Một số được gọi là "công bằng"nếu nó **chia hết cho tất cả các chữ số khác 0** của chính nó.

- **Ví dụ 1:** 120 là số công bằng vì $120 \vdots 1$ và $120 \vdots 2$ (số 0 bỏ qua).
- **Ví dụ 2:** 7 là số công bằng vì $7 \vdots 7$.
- **Ví dụ 3:** 23 **không** phải số công bằng vì 23 không chia hết cho 3.

Lộ trình tư duy (Micro-chunking)

Chúng ta sẽ chia bài toán thành 3 phần chính:

1. **Chunk 1:** Hiểu cách kiểm tra một số có "công bằng"hay không.
2. **Chunk 2:** Chiến thuật tìm số x nhỏ nhất (Duyệt hay dùng công thức?).
3. **Chunk 3:** Ước lượng giới hạn (Liệu tìm kiếm có quá lâu không?).

Chunk 1: Kiểm tra tính "Công bằng"

Để biết một số x có chia hết cho tất cả các chữ số của nó hay không:

1. **Tách từng chữ số** của x.
2. **Kiểm tra điều kiện:** Với mỗi chữ số $d \neq 0$, liệu $x \pmod{d} = 0$?

Ẩn dụ

Vị vua x chỉ "công bằng"nếu có thể chia đều ngân khố cho tất cả các quan đại thần (chữ số) của mình (ngoại trừ quan tên "Không").

Thử thách tư duy

Câu hỏi

Kiểm tra số 123. Các chữ số là: $1, 2, 3$.

- $123 \vdots 1$?

Thử thách tư duy

Câu hỏi

Kiểm tra số 123. Các chữ số là: $1, 2, 3$.

- $123 \vdots 1$? **Có**
- $123 \vdots 2$?

Thử thách tư duy

Câu hỏi

Kiểm tra số 123. Các chữ số là: $1, 2, 3$.

- $123 \vdots 1$? **Có**
- $123 \vdots 2$? **Không** (123 là số lẻ)
- $123 \vdots 3$?

Thử thách tư duy

Câu hỏi

Kiểm tra số 123. Các chữ số là: $1, 2, 3$.

- $123 \vdots 1$? **Có**
- $123 \vdots 2$? **Không** (123 là số lẻ)
- $123 \vdots 3$? **Có** ($1 + 2 + 3 = 6 \vdots 3$)

Thử thách tư duy

Câu hỏi

Kiểm tra số 123. Các chữ số là: $1, 2, 3$.

- $123 \vdots 1$? **Có**
- $123 \vdots 2$? **Không** (123 là số lẻ)
- $123 \vdots 3$? **Có** ($1 + 2 + 3 = 6 \vdots 3$)

Kết luận

Số 123 **không** phải là số công bằng vì không chia hết cho 2.

Chunk 2: Tại sao duyệt trâu (Brute Force) lại nhanh?

BCNN của các số từ $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ là:

$$2^3 \times 3^2 \times 5 \times 7 = 2520$$

Ý nghĩa con số 2520

Cứ trong khoảng 2520 số liên tiếp, **chắc chắn** sẽ có ít nhất một số chia hết cho tất cả các chữ số từ 1 đến 9.

Vì khoảng cách tối đa chỉ là 2520 (thực tế còn nhỏ hơn), việc dùng vòng lặp `while` để tăng dần n là cực kỳ hiệu quả!

Chunk 3: Tổng kết thuật toán & Mã giả

Thuật toán

1. Đọc số n.
2. Chừng nào n chưa phải số công bằng, tăng n lên 1.

```
1  bool kiem_tra_cong_bang(long long x) {
2      long long temp = x;
3      while (temp > 0) {
4          int d = temp % 10;
5          if (d != 0 && x % d != 0) {
6              return false;
7          }
8          temp /= 10;
9      }
10     return true;
11 }
12
```

Listing 1: Mã giả kiểm tra số công bằng

Thử thách cuối cùng

Câu hỏi

Nếu đề bài cho $n = 23$, máy tính sẽ thực hiện thế nào?

- **A.** Kiểm tra 23 (sai), tăng lên 24. Kiểm tra 24...
- **B.** Nhảy một phát đến 2520.

Thử thách cuối cùng

Câu hỏi

Nếu đề bài cho $n = 23$, máy tính sẽ thực hiện thế nào?

- **A.** Kiểm tra 23 (sai), tăng lên 24. Kiểm tra 24...
- **B.** Nhảy một phát đến 2520.

Đáp án: A. Kiểm tra 24: $24 \vdots 2$ và $24 \vdots 4$. **Dừng lại!** Chỉ mất 2 bước.

Bước cuối: Hiện thực hóa & Lưu ý

Đề bài cho $n \leq 10^{18}$.

Lưu ý quan trọng về kiểu dữ liệu

Bắt buộc dùng `long long` trong C++ để chứa n.

- `int`: Tối đa $\approx 2 \times 10^9$
- `long long`: Tối đa $\approx 9 \times 10^{18}$

Tóm tắt các bước

1. Đọc n kiểu `long long`.
2. Viết hàm `isFair(x)`.
3. Vòng lặp: `while(!isFair(n)) n++;`
4. In kết quả n.

Bạn muốn tiếp tục thế nào?

1. **Thử thách mới:** Một bài toán khác có tư duy tương tự.
2. **Nâng cấp:** Phân tích bài khó hơn (Quy hoạch động/Tham lam).
3. **Hỗ trợ Code:** Chuyển mã giả thành code C++ hoàn chỉnh.