

Giải quyết bài toán Ordinary Numbers

Codeforces 1520B

Algorithmic Coach

2026

Tiếp nhận & Phẫu thuật (Briefing)

Định nghĩa: Số bình thường

Một số được gọi là "bình thường" nếu **tất cả các chữ số của nó đều giống nhau**.

- **Ví dụ:** 1, 2, 11, 333 là số bình thường.
- **Phản ví dụ:** 12, 121 **không** phải là số bình thường.
- **Nhiệm vụ:** Cho số nguyên n , đếm số lượng số bình thường trong khoảng từ 1 đến n .

Lộ trình tư duy

- ❶ Nhận diện quy luật theo từng "bậc".
- ❷ Công thức hóa để tính nhanh với $n \leq 10^9$.
- ❸ Xây dựng thuật toán tối ưu.

Chunk 1: Quy luật của những "vị khách giống hệt nhau"

Hãy tưởng tượng các số bình thường theo từng nhóm chữ số:

- **Đoàn có 1 chữ số:** 1, 2, 3, 4, 5, 6, 7, 8, 9 (9 số).
- **Đoàn có 2 chữ số:** 11, 22, 33, 44, 55, 66, 77, 88, 99 (9 số).
- **Đoàn có 3 chữ số:** 111, 222, ..., 999 (9 số).

Bấy tư duy

Mỗi "bậc"(số chữ số) luôn chỉ có tối đa 9 số bình thường, dù n có lớn đến đâu.

Thử thách tư duy số 1

Câu hỏi

Nếu cho số $n = 45$, có bao nhiêu số bình thường từ 1 đến 45?

Thử thách tư duy số 1

Câu hỏi

Nếu cho số $n = 45$, có bao nhiêu số bình thường từ 1 đến 45?

Đáp án:

- Hàng đơn vị: 1, 2, 3, 4, 5, 6, 7, 8, 9 (9 số).
- Hàng chục: 11, 22, 33, 44 (4 số).
- **Tổng cộng: 13 số.**

Phát triển quy luật tổng quát

Giả sử xét $n = 715$:

- **Bậc 1 chữ số:** 1...9 (9 số).
- **Bậc 2 chữ số:** 11...99 (9 số).
- **Bậc 3 chữ số:** Các số ứng viên 111, 222, 333, 444, 555, 666, 777 ...

Thử thách

Với $n = 715$, số bình thường lớn nhất có 3 chữ số thỏa mãn là bao nhiêu?
Tổng cộng có bao nhiêu số từ 1...715?

Phát triển quy luật tổng quát

Giả sử xét $n = 715$:

- **Bậc 1 chữ số:** $1 \dots 9$ (9 số).
- **Bậc 2 chữ số:** $11 \dots 99$ (9 số).
- **Bậc 3 chữ số:** Các số ứng viên $111, 222, 333, 444, 555, 666, 777 \dots$

Thử thách

Với $n = 715$, số bình thường lớn nhất có 3 chữ số thỏa mãn là bao nhiêu?
Tổng cộng có bao nhiêu số từ $1 \dots 715$?

Giải đáp:

- Số lớn nhất là 666 (vì $777 > 715$).
- Tổng cộng: $9 + 9 + 6 = 24$ số.

Chunk 2: Công thức hóa "Siêu tốc"

Để xử lý n lên tới 10^9 :

- ① Đếm số lượng chữ số của n (gọi là k).
- ② Với mỗi bậc nhỏ hơn k , cộng thêm 9 số.
- ③ Với bậc cuối cùng (bậc có k chữ số), kiểm tra các số dạng $x, xx, xxx \dots$

Ví dụ với $n = 210$

- Bậc 1 chữ số: 9 số.
- Bậc 2 chữ số: 9 số.
- Bậc 3 chữ số: Chỉ có $111 \leq 210$ (1 số).
- **Tổng:** $9 + 9 + 1 = 19$ số.

Chunk 3: Xây dựng giải pháp tổng quát

Xét ví dụ $n = 332$:

- ① Số chữ số $k = 3$.
- ② Phần "chắc chắn có": $(3 - 1) \times 9 = 18$ số.
- ③ Độ có 3 chữ số: Kiểm tra các số $\{111, 222, 333\}$.

Câu hỏi

Trong các số $\{111, 222, 333\}$, số nào ≤ 332 ? Tổng kết quả là bao nhiêu?

Chunk 3: Xây dựng giải pháp tổng quát

Xét ví dụ $n = 332$:

- ① Số chữ số $k = 3$.
- ② Phần "chắc chắn có": $(3 - 1) \times 9 = 18$ số.
- ③ Độ có 3 chữ số: Kiểm tra các số $\{111, 222, 333\}$.

Câu hỏi

Trong các số $\{111, 222, 333\}$, số nào ≤ 332 ? Tổng kết quả là bao nhiêu?

Đáp án:

- Thỏa mãn: 111, 222 (2 số).
- Tổng cộng: $18 + 2 = 20$ số.

Chunk 4: Tổng kết thuật toán (Final Logic)

Mã giả (Pseudocode)

- ① Nhập n , khởi tạo $dem = 0$.
- ② Lặp chu_so từ 1 đến 9:
 - Khởi tạo $so_tao_ra = chu_so$.
 - Trong khi $so_tao_ra \leq n$:
 - Tăng dem .
 - $so_tao_ra = so_tao_ra * 10 + chu_so$.
- ③ In ra dem .

Ví dụ tạo số với $chu_so = 2$

$$2 \rightarrow (2 \times 10 + 2) = 22 \rightarrow (22 \times 10 + 2) = 222 \dots$$

Kết luận

- Cách tiếp cận này giúp "quét" sạch mọi số bình thường mà không cần quan tâm n lớn hay nhỏ.
- Độ phức tạp cực thấp: Chỉ khoảng 9×9 lần lặp cho $n = 10^9$.

Bước tiếp theo

Bạn có muốn xem mã nguồn chi tiết bằng C++ hay Python không, hay chúng ta sẽ thử sức với một bài toán mới?