Huấn luyện viên Tư duy Thuật toán

Phân tích bài toán Extremely Round (1766A)

Slide Learning C++

Ngày 21 tháng 1 năm 2026

Giới thiệu

- Chào mừng bạn đến với chương trình huấn luyện tư duy thuật toán.
- Chúng ta sẽ "mổ xẻ"bài toán theo triết lý **Micro-chunking** để nắm chắc gốc rễ vấn đề.

Mục tiêu

Giải quyết bài toán đếm số "cực tròn"một cách tối ưu.

Bước 1: Phẫu thuật đề bài (Deconstruct)

Định nghĩa: Số "Extremely Round"

Là số nguyên dương mà trong các chữ số của nó, **chỉ có duy nhất một chữ số khác 0**.

Ví dụ

- **Đúng:** 5, 400, 9000.
- **Sai:** 11, 405, 9100 (có từ 2 chữ số khác 0 trở lên).

Nhiệm vụ

Cho số nguyên n, đếm số lượng số "cực tròn"từ 1 đến n ($n \leq 999,999$).

Lộ trình tư duy

Chúng ta sẽ đi qua 3 mảnh kiến thức (Chunks):

1. **Chunk 1:** Nhận diện quy luật của các số "cực tròn".
2. **Chunk 2:** Cách đếm thông minh không cần duyệt từng số.
3. **Chunk 3:** Xử lý các bẫy logic và rút ra công thức.

Chunk 1: Truy tìm quy luật "Cực tròn"

Hãy tưởng tượng mỗi hàng (đơn vị, chục, trăm...) là một "ngăn tủ":

- **Hàng 1 chữ số:** 1, 2, ..., 9 (9 số).
- **Hàng 2 chữ số:** 10, 20, ..., 90 (9 số).
- **Hàng 3 chữ số:** 100, 200, ..., 900 (9 số).

Thử thách tư duy

Nếu $n = 35$, có bao nhiêu số cực tròn nhỏ hơn hoặc bằng 35?

Chunk 1: Truy tìm quy luật "Cực tròn"

Hãy tưởng tượng mỗi hàng (đơn vị, chục, trăm...) là một "ngăn tủ":

- **Hàng 1 chữ số:** 1, 2, ..., 9 (9 số).
- **Hàng 2 chữ số:** 10, 20, ..., 90 (9 số).
- **Hàng 3 chữ số:** 100, 200, ..., 900 (9 số).

Thử thách tư duy

Nếu $n = 35$, có bao nhiêu số cực tròn nhỏ hơn hoặc bằng 35?

- Hàng đơn vị: 1, 2, 3, 4, 5, 6, 7, 8, 9 (9 số)
- Hàng chục: 10, 20, 30 (3 số)
- **Tổng cộng: 12 số.**

Chunk 2: Tổng quát hóa quy luật

Với mỗi "bậc"(chữ số), ta luôn có tối đa **9 số** cực tròn. Để đếm đến n, ta chia làm 2 phần:

1. **Phần nguyên:** Những "bậc"đã đi qua hết hoàn toàn.
2. **Phần dư:** Những số cực tròn ở bậc hiện tại của n.

Ví dụ với $n = 4567$

- Bậc đơn vị, chục, trăm: $3 \times 9 = 27$ số.
- Bậc nghìn (hiện tại): 1000, 2000, 3000, 4000 (4 số).
- **Tổng cộng:** $27 + 4 = 31$ số.

Chunk 3: Rút ra công thức

Gọi s là số chữ số của n, và d là chữ số đầu tiên của n.

Mối liên hệ

- Số bậc đầy đủ là $(s-1)$.
- Mỗi bậc đầy đủ có 9 số cực tròn.
- Bậc cuối cùng có đúng d số.

Công thức tổng quát

$$\text{Result} = (s-1) \times 9 + d$$

Thử nghiệm giới hạn

Kiểm tra với $n = 999,999$

- Số chữ số $s = 6$.
- Chữ số đầu tiên $d = 9$.
- Áp dụng: $(6-1) \times 9 + 9 = 5 \times 9 + 9 = 54$.

Giải thích: Có 6 bậc (từ đơn vị đến trăm nghìn), mỗi bậc đóng góp 9 số.

Bước 3: Hiện thực hóa thành thuật toán

Cách lấy d và s bằng toán học

Sử dụng biến tạm go_n để không làm mất giá trị gốc của n.

```
Nhap t (so bo test)
Lap lai t lan:
    Nhap n
    go_n = n
    s = do_dai_chuoi(n) # Hoac dem trong khi chia 10

    # Tim chu so dau tien d
    Trong khi go_n >= 10:
        go_n = go_n // 10
    d = go_n

    In ra (s - 1) * 9 + d
```

Listing 1: Mã giả logic xử lý

Thử thách thực hành

Câu hỏi suy luận

Nếu $n = 7$ (số có 1 chữ số):

- Vòng lặp `while (go_n >= 10)` có chạy không?
- Giá trị d cuối cùng là bao nhiêu?
- Công thức $(s - 1) \times 9 + d$ có trả về đúng 7 không?

Thử thách thực hành

Câu hỏi suy luận

Nếu $n = 7$ (số có 1 chữ số):

- Vòng lặp `while (go_n >= 10)` có chạy không?
- Giá trị d cuối cùng là bao nhiêu?
- Công thức $(s - 1) \times 9 + d$ có trả về đúng 7 không?

Đáp án

- Vòng lặp không chạy. $d = 7$.
- $s = 1 \Rightarrow (1 - 1) \times 9 + 7 = 7$.
- **Kết luận:** Thuật toán chạy đúng cho cả số có 1 chữ số!