

Giải Mã Tư Duy Thuật Toán

Bài toán: Codeforces 151A - Soft Drinking

Coach Tư Duy Thuật Toán (LHTL Edition)

Ngày 13 tháng 1 năm 2026

Nội dung bài học

Codeforces 151A - Soft Drinking

Chào mừng bạn!

Chúng ta sẽ cùng nhau "mở xẻ" bài **Codeforces 151A - Soft Drinking**.

Đây là một bài toán kinh điển về:

- **Quản lý tài nguyên.**
- **Tìm điểm thắt nút (Bottleneck).**

Đừng để đồng biến số n, k, l, c, d, p, nl, np làm bạn hoa mắt. Chúng ta sẽ gọn gàng nó ngay bây giờ.

1. Phẫu thuật đề bài (Deconstruct)

Hãy quên chuyện "uống nước ngọt" đi. Hãy tưởng tượng bạn là một **Bar trưởng** đang pha chế các **Combo đồ uống**.

Quy tắc pha 1 Combo

Bạn **BẮT BUỘC** phải có đủ 3 thành phần cùng lúc:

- ① **Nước**: Một lượng ml nhất định.
- ② **Chanh**: Một lát chanh.
- ③ **Muối**: Một lượng gam muối.

Nếu thiếu bất kỳ cái nào → Không thể tạo ra Combo.

Dữ liệu đầu vào (Kho hàng)

Dữ liệu đầu vào (Input) thực chất là kho hàng của bạn:

- **Nhân lực:** n (số người bạn).
- **Kho Nước:** k chai, mỗi chai $/ ml$.
- **Kho Chanh:** c quả, mỗi quả cắt được d lát.
- **Kho Muối:** p gam.
- **Công thức pha 1 Combo:** Cần n/l ml nước và np gam muối (và luôn luôn là 1 lát chanh).

2. Lộ trình tư duy

Chúng ta sẽ đi qua 3 mảnh ghép (Chunks):

- ① **Chunk 1: Tổng kiểm kê hàng**
(Quy đổi mọi thứ về đơn vị nhỏ nhất).
- ② **Chunk 2: Quy tắc "Chiếc thùng gỗ" (Bottleneck)**
(Tìm xem nguyên liệu nào sẽ hết trước).
- ③ **Chunk 3: Chia phần công bằng**
(Tính ra kết quả cuối cùng cho mỗi người).

Chunk 1: Tổng kiểm kê kho hàng

Trước khi pha chế, bạn phải biết trong kho mình có **tổng cộng** bao nhiêu nguyên liệu rồi.

Quy đổi đơn vị

- **Nước:** Bạn có k chai, mỗi chai $/ \text{ml}$.
- **Chanh:** Bạn có c quả, mỗi quả cắt được d lát.
- **Muối:** Đã có sẵn p gam (không cần tính).

Nhiệm vụ: Tính tổng số lượng thực tế có thể dùng được.

Thử thách tư duy (Mental Check)

Giả sử kho hàng có số liệu sau:

- $k = 3$ (chai), $l = 100$ (ml/chai).
- $c = 5$ (quả chanh), $d = 2$ (lát/quả).
- $p = 50$ (gam muối).

Kết quả kiểm kê:

- ① Nước: $3 \times 100 = 300$ ml.
- ② Chanh: $5 \times 2 = 10$ lát.
- ③ Muối: 50 gam.

Lưu ý quan trọng

Ở bước này, muối là p , không phải p/np .
Phép chia thuộc về bước "Tính khả năng phục vụ" sau này.

Chunk 2: Quy tắc "Chiếc thùng gỗ"(Bottleneck)

Bây giờ bạn đã có tổng nguyên liệu. Chúng ta cần xem từng loại nguyên liệu **độc lập** có thể tạo ra tối đa bao nhiêu Combo.

Công thức pha 1 Combo cần: n_l ml nước, 1 lát chanh, n_p gam muối.

Nguyên lý Chiếc thùng gỗ

Bạn chỉ có thể tạo ra số Combo bằng với nguyên liệu **ít nhất/hạn hẹp nhất**.
(Giống như thùng gỗ chỉ đựng được nước đến chiều cao của thanh gỗ thấp nhất).

Thử thách tư duy (Quan trọng)

Dữ liệu:

- Kho: Nước **300**, Chanh **10**, Muối **50**.
- Công thức cần: **5** ml nước, **1** lát chanh, **2** gam muối.

Tính toán:

- ① Nước: $300/5 = 60$ ly.
- ② Chanh: $10/1 = 10$ ly.
- ③ Muối: $50/2 = 25$ ly.

Quy luật cốt lõi: Dù nước đủ 60 ly, muối đủ 25 ly, nhưng làm đến ly thứ 10 thì chanh hết sạch!
→ Số Combo tối đa = **$\min(60, 10, 25) = 10$ Combo.**

Chunk 3: Chia phần công bằng (Final Step)

Chúng ta đã có **10 Combo** trên bàn.

Đề bài hỏi: "**Mỗi người bạn nhận được bao nhiêu ly?**"

Dữ kiện cuối cùng: Số người bạn n .

Quy tắc chia

Phải chia đều. Nếu chia không hết thì bỏ phần dư (chia lấy phần nguyên), vì không thể uống "nửa ly" trong bài toán này.

Thử thách về đích

Giả sử có tổng **10 Combo**.

TH1: $n = 3$ người

$$10/3 = 3$$

(Dư 1, bỏ qua)

TH2: $n = 4$ người

$$10/4 = 2$$

(Dư 2, bỏ qua)

TH3: $n = 11$ người

$$10/11 = 0$$

(Không đủ chia)

Số nguyên chia nhau sẽ mất phần thập phân (trong C++/Python/Java).

Tổng kết chiến thuật (Wrap-up)

Đây là **Bản thiết kế thuật toán** cho bài 151A:

① Bước 1: Quy đổi tổng lực

- $Total_Water = k \times l$
- $Total_Lime = c \times d$
- $Total_Salt = p$

② Bước 2: Tìm giới hạn (Bottleneck)

- $Toast_Water = Total_Water/nl$
- $Toast_Lime = Total_Lime/1$
- $Toast_Salt = Total_Salt/np$
- $\Rightarrow Max_Toast = \min(Toast_Water, Toast_Lime, Toast_Salt)$

③ Bước 3: Chia phần

- $Result = Max_Toast/n$

Giao nhiệm vụ cuối

Bây giờ là lúc bạn chuyển tư duy này thành code.

Gợi ý khi code

Hàm tìm giá trị nhỏ nhất:

- **C++:**

```
min({a, b, c})  
// Hoặc  
min(a, min(b, c))
```

- **Python:**

```
min(a, b, c)
```

Hãy thử viết code và nộp bài (submit). Nếu gặp lỗi, hãy quay lại đây để "debug" tư duy!

Chúc bạn thành công!

Hẹn gặp lại ở bài toán tiếp theo.