

Phân tích Tư duy Thuật toán: Codeforces 732A

Triết lý "Learning How to Learn"

Coach Tư Duy Thuật Toán

Ngày 13 tháng 1 năm 2026

Lời mở đầu

Chào mừng bạn! Tôi đã nhận tín hiệu. Tôi là **Coach Tư Duy Thuật Toán** của bạn đây.
Hôm nay chúng ta sẽ cùng "mổ xẻ" bài toán **Codeforces 732A - Buy a Shovel (Mua xẻng)**.

Lưu ý

Đừng lo về code vội, hãy cất bàn phím đi và dùng "bộ não" trước đã.

Bước 1: Tiếp nhận & Phẫu thuật (Briefing)

Đề bài kể về anh chàng Polycarp đi mua xăng, nhưng chúng ta hãy gạt bỏ câu chuyện đó sang một bên. Đây là cốt lõi logic của vấn đề:

- **1. Dữ liệu đầu vào (Input):**

- k : Giá tiền của **một** chiếc xăng.
- r : Mệnh giá của một đồng xu lẻ duy nhất mà bạn có (trong túi bạn có vô hạn tờ tiền mệnh giá 10, nhưng chỉ có đúng **một** đồng xu mệnh giá r).

- **2. Mục tiêu (Goal):**

- Tìm số lượng xăng **ít nhất** (gọi là x) mà bạn cần mua.
- **Điều kiện:** Bạn phải trả tiền **vừa đủ khít** (không thối lại tiền thừa).

- **3. Công cụ thanh toán:**

- Bạn chỉ có thể trả bằng các tờ tiền **10 đồng** (số lượng vô hạn).
- Và (tùy chọn) dùng thêm đúng **một đồng xu** r nếu muốn.

Lộ trình tư duy (Roadmap)

Chúng ta sẽ giải quyết bài toán này qua 3 mảnh ghép (Chunks):

- ① **Chunk 1:** Phân tích "Cơ chế ví tiền" (Điều kiện chia hết).
- ② **Chunk 2:** Truy tìm "Chữ số tận cùng" (Last Digit Logic).
- ③ **Chunk 3:** Chiến thuật thử sai (Simulation).

Bạn đã sẵn sàng bước vào Chunk 1: Phân tích cơ chế ví tiền chưa?

Chunk 1: Cơ chế ví tiền (Điều kiện thanh toán)

Hãy tưởng tượng bạn đang đứng trước quầy thu ngân. Tổng số tiền bạn phải trả là:

$$S = k \times x$$

(Trong đó: k là giá 1 cái xéng, x là số xéng bạn mua).

Trong túi bạn có:

- 1 Rất nhiều tờ **10 đồng**.
- 2 Đúng **1 đồng xu lẻ** mệnh giá r .

Để trả vừa đủ, tổng số tiền S phải thỏa mãn 1 trong 2 trường hợp:

- **Trường hợp 1:** Dùng toàn tờ 10 đồng $\rightarrow S$ chia hết cho 10 (Tận cùng là **0**).
- **Trường hợp 2:** Dùng tờ 10 đồng + đồng xu $r \rightarrow (S - r)$ chia hết cho 10 (Tận cùng là r).

TÓM LẠI: Chữ số tận cùng của S bắt buộc phải là **0** hoặc r .

Chunk 1: Cạm bẫy & Thử thách tư duy

Cạm bẫy (Trap)

Nhiều bạn nghĩ rằng bắt buộc phải dùng đồng xu r . **Sai!** Bạn có thể cắt đồng xu đó đi nếu Tổng tiền S đã tròn chục (tận cùng là 0).

Thử thách tư duy (Mental Check)

Giả sử giá xăng $k = 12$ và đồng xu lẻ $r = 8$. Hãy kiểm tra:

- 1 Mua $x = 1 \rightarrow S = 12$ (Tận cùng 2). Có trả được không? **Không**.
- 2 Mua $x = 4 \rightarrow S = 48$ (Tận cùng 8). Có trả được không? **Có** (trùng $r = 8$).
- 3 Mua $x = 5 \rightarrow S = 60$ (Tận cùng 0). Có trả được không? **Có** (tròn chục).

Bạn đã nắm vững "Điều kiện thắng"!

Chunk 2: Chiến thuật truy tìm (The Loop)

Đề bài yêu cầu tìm số xẻng **ít nhất**. Cách đơn giản nhất là "**Thử từng cái một**".

Giống như bạn đi thử chìa khóa vào ổ vậy:

- 1 Thử mua 1 cái ($x = 1$) \rightarrow Tính tổng tiền \rightarrow Kiểm tra đũa.
 - Nếu đúng: Dừng lại ngay! (Đây là đáp án nhỏ nhất).
 - Nếu sai: Thử tiếp 2 cái ($x = 2$).
- 2 Lặp lại cho đến khi tìm thấy.

Để máy tính "nhìn" được chữ số tận cùng, dùng phép toán **Chia lấy dư cho 10** (Modulus % 10).

- Ví dụ: $48 \% 10 = 8$
- Ví dụ: $60 \% 10 = 0$

Chunk 2: Cạm bẫy & Thử thách tư duy

Cạm bẫy (Trap)

"Nhỡ thử mãi không tìm được thì sao? Nhỡ nó chạy đến vô tận thì sao?"

Yên tâm: Vì chỉ quan tâm chữ số tận cùng (0-9), quy luật sẽ lặp lại.
Thực tế chỉ cần thử tối đa 10 lần.

Thử thách tư duy (Mental Check)

Máy tính chạy bằng cơ. $k = 7$, $r = 3$. Thử dần x tăng lên:

- $x = 1 \rightarrow S = 7$ (Sai: $7 \neq 0, 7 \neq 3$)
- $x = 2 \rightarrow S = 14$ (Sai)
- ... (Bạn đã tính ra) ...
- $x = 9 \rightarrow S = 63$ (Tận cùng là 3 \rightarrow Trùng $r = 3$).

Bingo! Chúng ta dừng lại và mua 9 cái.

Chunk 3: Tổng kết & Mã giả (Wrap Up)

Thuật toán (Algorithm):

1. Nhập k, r .
2. Vòng lặp đếm i từ 1 tăng dần mãi.
3. Mỗi bước tính 'Tổng tiền = $k * i$ '.
4. Kiểm tra: Nếu ('

Mã giả (Pseudocode):

```
c      k, r

Cho i c h y t 1 n v t n :
    S = k * i
    duoi = S % 10 // L y c h s t n c n g

    N u (duoi == 0) H 0 C (duoi == r):
        In ra i
        D n g v n g l p (Break)
```

Bài tập về đích (Final Check)

Trước khi tôi đưa bạn xem code C++/Python mẫu, hãy xác nhận một điều nhỏ về cú pháp logic:

Câu hỏi

Trong lập trình, điều kiện **"Hoặc"** được viết là gì?

- ☐ A &&
- ☐ B ||

Chọn đáp án đúng để mở khóa Code mẫu nhé!