

Codeforces 546A - Soldier and Bananas

Tư duy thuật toán & Cấp số cộng

Coach Tư duy thuật toán

Ngày 14 tháng 1 năm 2026

Giới thiệu

Lời chào từ Coach

Chào mừng bạn. Tôi là Coach Tư duy thuật toán của bạn đây.
Đã nhận yêu cầu: **Codeforces 546A - Soldier and Bananas.**

- **Đánh giá:** Bài toán nhập môn kinh điển (Rate 800).
- **Kiến thức cốt lõi:**
 - Cấp số cộng (Arithmetic Progression).
 - Xử lý số âm (Negative handling).

Chúng ta sẽ không code vội. Hãy mổ xẻ nó trước.

BƯỚC 1: TIẾP NHẬN & PHẪU THUẬT

Tóm tắt đề bài (Ngôn ngữ con người)

Bài toán này thực chất là một bài toán đi chợ với quy luật giá tăng dần:

- Bạn muốn mua w quả chuối.
- Quả thứ 1 giá: $1k$ đô.
- Quả thứ 2 giá: $2k$ đô.
- Quả thứ 3 giá: $3k$ đô.
- ... Cứ thế tăng lên theo cấp số nhân với k .
- Trong túi bạn đang có sẵn n đô.

Mục tiêu: Tính số tiền bạn **phải vay thêm** để mua đủ w quả.

Các nhân vật chính (Input)

- k : Giá gốc (hệ số giá).
- n : Tiền có sẵn (ví tiền).
- w : Số lượng cần mua.

Lộ trình tư duy (Roadmap): Để giải quyết bài này, chúng ta cần đi qua 3 mảnh ghép (Chunks):

- ① **Chunk 1:** Hiểu quy luật tính tiền (Tư duy quy nạp).
- ② **Chunk 2:** Tính tổng hóa đơn (Tư duy tối ưu toán học).
- ③ **Chunk 3:** Tính số tiền nợ & Bẫy Logic (Edge Cases).

CHUNK 1: QUY LUẬT "CẦU THANG GIÁ CẢ"

Mô hình Cầu thang

Giá chuỗi không cố định mà tăng dần như cầu thang:

- Bậc 1 (Quả thứ 1): Giá gốc là 1 lần k .
- Bậc 2 (Quả thứ 2): Giá tăng lên 2 lần k .
- Bậc 3 (Quả thứ 3): Giá tăng lên 3 lần k .
- ...
- Bậc i (Quả thứ i): Giá là $i \times k$.

Càng mua nhiều, giá từng quả càng đắt đỏ.

THỦ THÁCH TƯ DUY (MENTAL CHECK)

Câu hỏi

Giả sử hệ số giá $k = 3$.

Tôi muốn bạn tính giá tiền của **riêng quả chuối thứ 4** (chỉ quả đó thôi, không phải tổng).

- A. 3 đô
- B. 9 đô
- C. 12 đô

THỦ THÁCH TƯ DUY (MENTAL CHECK)

Câu hỏi

Giả sử hệ số giá $k = 3$.

Tôi muốn bạn tính giá tiền của **riêng quả chuối thứ 4** (chỉ quả đó thôi, không phải tổng).

- A. 3 đô
- B. 9 đô
- C. 12 đô

Đáp án: C

Chính xác! Giá quả thứ 4 là $4 \times 3 = 12$.

Công thức: Giá quả thứ $i = i \times k$.

CHUNK 2: TÍNH TỔNG HÓA ĐƠN (SUMMATION)

Công thức tính tổng tiền cho w quả:

$$Total = 1k + 2k + 3k + \cdots + wk$$

Nhìn kỹ nhé (Pattern Recognition)

Nếu ta đặt k ra ngoài làm nhân tử chung, ta sẽ có:

$$Total = k \times (1 + 2 + 3 + \cdots + w)$$

Cụm $(1 + 2 + \cdots + w)$ là tổng các số tự nhiên liên tiếp.

- *Trong toán học:* Đây là cấp số cộng.
- *Trong lập trình:* Dùng vòng lặp hoặc công thức $\frac{w(w+1)}{2}$.

THỦ THÁCH TƯ DUY (MENTAL CHECK)

Câu hỏi

Vẫn giữ $k = 3$. Lần này bạn muốn mua **4 quả** ($w = 4$).

Hãy tính **Tổng số tiền** bạn phải trả cho cả 4 quả này là bao nhiêu?

- A. 12
- B. 30
- C. 24

THỦ THÁCH TƯ DUY (MENTAL CHECK)

Câu hỏi

Vẫn giữ $k = 3$. Lần này bạn muốn mua **4 quả** ($w = 4$).

Hãy tính **Tổng số tiền** bạn phải trả cho cả 4 quả này là bao nhiêu?

- A. 12
- B. 30
- C. 24

Dáp án: B

Quá chuẩn! Bạn đã cộng dồn rất đúng:

$$3 \times (1 + 2 + 3 + 4) = 3 \times 10 = 30.$$

CHUNK 3: ĐIỀU KIỆN VAY MƯỢN & CÁI BẤY SỐ ÂM

Logic thông thường:

$$\text{Tiền thiếu} = \text{Tổng chi phí} - \text{Tiền đang có}$$

$$\text{Result} = Total - n$$

Nhưng cuộc đời không phải lúc nào cũng cần vay mượn.

Tình huống thực tế

Nếu tiền trong túi (n) lớn hơn hóa đơn ($Total$), bạn không cần vay ai cả.
Số tiền nợ phải là 0, chứ không phải số âm.

THỦ THÁCH TƯ DUY (TRAP CHECK)

Câu hỏi

Giả sử hóa đơn là **30 đô**. Bạn có **40 đô** trong ví ($n = 40$). Máy tính cần in ra (số tiền phải vay) là bao nhiêu?

- A. -10 (Vì $30 - 40 = -10$)
- B. 0 (Không cần vay gì cả)
- C. 10 (Vẫn vay cho chắc?)

THỦ THÁCH TƯ DUY (TRAP CHECK)

Câu hỏi

Giả sử hóa đơn là **30 đô**. Bạn có **40 đô** trong ví ($n = 40$). Máy tính cần in ra (số tiền phải vay) là bao nhiêu?

- A. -10 (Vì $30 - 40 = -10$)
- B. 0 (Không cần vay gì cả)
- C. 10 (Vẫn vay cho chắc?)

Dáp án: B

Tuyệt vời! Bạn đã tránh được cái "bẫy" phổ biến nhất.

Tại sao lại là 0? Không ai nói "Tôi phải vay âm 10 đô" cả. Nếu $Total \leq n$, kết quả là 0.

TỔNG KẾT CHIẾN THUẬT (THE ALGORITHM)

① Nhập dữ liệu (Input): Lấy 3 số k, n, w .

② Tính Tổng hóa đơn (Total Cost):

- Cách 1 (Loop): Vòng lặp từ 1 đến w , cộng dồn.
- Cách 2 (Math - Khuyên dùng):

$$Total = k \times \frac{w \times (w + 1)}{2}$$

③ Tính tiền nợ (Result):

- Tính độ lệch: $diff = Total - n$.

• Quyết định (Decision):

- Nếu $diff > 0 \rightarrow$ In ra $diff$.
- Nếu $diff \leq 0 \rightarrow$ In ra 0.

Công thức Pro (One-liner)

$$\text{Result} = \max(0, Total - n)$$

BƯỚC TIẾP THEO

Hành động ngay

Bây giờ bạn đã có đủ tư duy logic trong đầu.

Nhiệm vụ của bạn: Hãy viết code (C++, Python hoặc ngôn ngữ bạn chọn) và nộp thử.

Bạn có muốn dán đoạn code của bạn vào đây để tôi review không?