

Codeforces 116A - Tram

Rèn luyện tư duy Mô phỏng (Simulation)

Slide Learning Cpp

Ngày 14 tháng 1 năm 2026

Chào mừng bạn đến với Codeforces 116A - Tram!

Đây là một bài toán kinh điển để rèn luyện tư duy **Mô phỏng (Simulation)** – tức là bắt máy tính làm theo đúng những gì diễn ra trong thực tế.

Chúng ta sẽ không vội viết code. Hãy cùng nhau "mổ xẻ" chiếc xe điện này nhé.

Bước 1: Phân thuật đề bài (Deconstruction)

1. Cốt truyện đời thường

Hãy tưởng tượng bạn là người quản lý một tuyến xe điện (Tram). Tuyến xe này đi qua n trạm dừng. Tại mỗi trạm, quy trình luôn diễn ra theo thứ tự:

- ❶ Cửa mở: Một số khách **xuống xe** (gọi là a_i).
- ❷ Sau đó: Một số khách **lên xe** (gọi là b_i).

Bước 1: Phẫu thuật đề bài (Deconstruction)

1. Cốt truyện đời thường

Hãy tưởng tượng bạn là người quản lý một tuyến xe điện (Tram). Tuyến xe này đi qua n trạm dừng. Tại mỗi trạm, quy trình luôn diễn ra theo thứ tự:

- ❶ Cửa mở: Một số khách **xuống xe** (gọi là a_i).
- ❷ Sau đó: Một số khách **lên xe** (gọi là b_i).

2. Mục tiêu

Bạn cần mua một chiếc xe có sức chứa **vừa đủ nhỏ nhất** nhưng vẫn đảm bảo **không bao giờ** bị quá tải.

Nói cách khác: Tìm con số **lớn nhất** (Maximum) lượng hành khách có mặt trên xe tại bất kỳ thời điểm nào trong suốt hành trình.

Chúng ta sẽ đi qua 3 bước (Chunks):

- **Chunk 1:** Hiểu cơ chế "Dòng chảy"(Ra và Vào).
- **Chunk 2:** Theo dõi "Biến động"(Số khách hiện tại).
- **Chunk 3:** Chốt hạ "Đỉnh điểm"(Kết quả).

Chunk 1: Cơ chế dòng chảy (Flow)

Trước khi tính toán cả hành trình, ta phải hiểu chuyện gì xảy ra tại **một trạm**.

Ấn dụ: Chiếc hồ nước

- Số người xuống (a_i): Là nước bị rút ra.
- Số người lên (b_i): Là nước được bơm thêm vào.

Quy tắc bắt di bắt dịch

Xe đến trạm \rightarrow Khách xuống trước \rightarrow Khách lên sau \rightarrow Xe chạy tiếp.

Lưu ý: Lúc khởi hành (trước trạm 1), xe luôn rỗng (0 người).

Thử thách tư duy (Mental Check)

Giả sử xe đang chạy và bên trong có **10 hành khách**. Xe đến trạm kế tiếp với dữ liệu:

- $a = 4$ (4 người xuống).
- $b = 7$ (7 người lên).

Câu hỏi: Sau khi xe rời trạm này, trên xe có bao nhiêu người?

Thử thách tư duy (Mental Check)

Giả sử xe đang chạy và bên trong có **10 hành khách**. Xe đến trạm kế tiếp với dữ liệu:

- $a = 4$ (4 người xuống).
- $b = 7$ (7 người lên).

Câu hỏi: Sau khi xe rời trạm này, trên xe có bao nhiêu người?

Đáp án:

Thử thách tư duy (Mental Check)

Giả sử xe đang chạy và bên trong có **10 hành khách**. Xe đến trạm kế tiếp với dữ liệu:

- $a = 4$ (4 người xuống).
- $b = 7$ (7 người lên).

Câu hỏi: Sau khi xe rời trạm này, trên xe có bao nhiêu người?

Đáp án:

Nếu bạn nghĩ là $7 - 4 = 3$... **Rất tiếc, chưa chính xác!** Đây chỉ là lượng khách *tăng thêm*.

Thử thách tư duy (Mental Check)

Giả sử xe đang chạy và bên trong có **10 hành khách**. Xe đến trạm kế tiếp với dữ liệu:

- $a = 4$ (4 người xuống).
- $b = 7$ (7 người lên).

Câu hỏi: Sau khi xe rời trạm này, trên xe có bao nhiêu người?

Đáp án:

Nếu bạn nghĩ là $7 - 4 = 3$... **Rất tiếc, chưa chính xác!** Đây chỉ là lượng khách *tăng thêm*.

Quy trình đúng

- 1 Trên xe có sẵn: **10 người**.
- 2 4 người xuống: $10 - 4 = 6$ người.
- 3 7 người lên: $6 + 7 = 13$ người.

Công thức cốt lõi: $New = Old - a + b$

Chunk 2: Theo dõi biến động (Accumulation)

Bài toán là cả một hành trình dài. Ta cần một biến để lưu trữ "**Số khách hiện tại**".

Giả sử biến đó tên là: `current_passengers`.

Quy luật quan trọng

- Số khách hiện tại của trạm này sẽ là **số khách khởi điểm** cho trạm kế tiếp.
- Giá trị `current_passengers` sẽ thay đổi liên tục qua từng trạm.

Thử thách liên hoàn (Chain Challenge)

Tuyến xe đi qua 3 trạm. Ban đầu xe rỗng (0 người).

Dữ liệu hành trình:

- ① **Trạm 1:** Xuống 0, Lên 5 ($0 - 0 + 5$)
- ② **Trạm 2:** Xuống 2, Lên 4 ($a = 2, b = 4$)
- ③ **Trạm 3:** ...

Câu hỏi: Ngay khi xe **rời khỏi Trạm 2**, trên xe có bao nhiêu người? (Hãy tính cộng dồn từ Trạm 1).

Thử thách liên hoàn (Chain Challenge)

Tuyến xe đi qua 3 trạm. Ban đầu xe rỗng (0 người).

Dữ liệu hành trình:

- ① **Trạm 1:** Xuống 0, Lên 5 ($0 - 0 + 5$)
- ② **Trạm 2:** Xuống 2, Lên 4 ($a = 2, b = 4$)
- ③ **Trạm 3:** ...

Câu hỏi: Ngay khi xe **rời khỏi Trạm 2**, trên xe có bao nhiêu người? (Hãy tính cộng dồn từ Trạm 1).

Đáp án

- **Rời Trạm 1:** $0 - 0 + 5 = 5$ khách.
- **Rời Trạm 2:** $5 - 2 + 4 = 7$ khách.

Chunk 3: Đi tìm đỉnh điểm (The Peak)

Câu hỏi chính: "**Sức chứa nhỏ nhất là bao nhiêu?**"

Tư duy

Để không ai bị bỏ lại, chiếc xe phải chứa được lượng khách vào lúc **đông nhất**.

Nhiệm vụ: Dùng một biến "camera" (gọi là `max_capacity`) để chụp lại con số lớn nhất từng xuất hiện.

- Ví dụ Trạm 1: 5 người → Kỷ lục: 5.
- Ví dụ Trạm 2: 7 người → Kỷ lục mới: 7 (vì $7 > 5$).
- Ví dụ Trạm 3: 2 người → Kỷ lục vẫn là 7 (vì $2 < 7$).

⇒ Kết quả cuối cùng là **7**.

Thử thách cuối cùng (Final Boss)

Có 4 trạm. Hãy điền vào (A) và (B) và tìm Đáp án cuối cùng.

Trạm	Xuống (a)	Lên (b)	Số khách sau khi rời trạm
1	0	3	3
2	2	5	? (A)
3	4	2	? (B)
4	4	0	0 (Hết chuyến)

Thử thách cuối cùng (Final Boss)

Có 4 trạm. Hãy điền vào (A) và (B) và tìm Đáp án cuối cùng.

Trạm	Xuống (a)	Lên (b)	Số khách sau khi rời trạm
1	0	3	3
2	2	5	? (A)
3	4	2	? (B)
4	4	0	0 (Hết chuyến)

Giải mã:

- Tại (A): $3 - 2 + 5 = 6$

Thử thách cuối cùng (Final Boss)

Có 4 trạm. Hãy điền vào (A) và (B) và tìm Đáp án cuối cùng.

Trạm	Xuống (a)	Lên (b)	Số khách sau khi rời trạm
1	0	3	3
2	2	5	? (A)
3	4	2	? (B)
4	4	0	0 (Hết chuyến)

Giải mã:

- Tại (A): $3 - 2 + 5 = 6$
- Tại (B): $6 - 4 + 2 = 4$

Thử thách cuối cùng (Final Boss)

Có 4 trạm. Hãy điền vào (A) và (B) và tìm Đáp án cuối cùng.

Trạm	Xuống (a)	Lên (b)	Số khách sau khi rời trạm
1	0	3	3
2	2	5	? (A)
3	4	2	? (B)
4	4	0	0 (Hết chuyến)

Giải mã:

- Tại (A): $3 - 2 + 5 = 6$
- Tại (B): $6 - 4 + 2 = 4$
- So sánh các thời điểm: 3, **6**, 4.

Thử thách cuối cùng (Final Boss)

Có 4 trạm. Hãy điền vào (A) và (B) và tìm Đáp án cuối cùng.

Trạm	Xuống (a)	Lên (b)	Số khách sau khi rời trạm
1	0	3	3
2	2	5	? (A)
3	4	2	? (B)
4	4	0	0 (Hết chuyến)

Giải mã:

- Tại (A): $3 - 2 + 5 = 6$
- Tại (B): $6 - 4 + 2 = 4$
- So sánh các thời điểm: 3, **6**, 4.

Kết quả

Sức chứa tối thiểu cần mua là **6**.

Tổng kết thuật toán (Blueprint)

Bản thiết kế (Mã giả - Pseudocode):

1. Chuẩn bị (Setup)

- Đọc số lượng trạm n .
- $hien_tai = 0$ (số khách đang ngồi trên xe).
- $suc_chua = 0$ (kỷ lục đồng nhất).

2. Vòng lặp hành trình (Loop)

Lặp n lần (đi qua từng trạm):

- Đọc 2 số a (xuống) và b (lên).
- **Cập nhật:** $hien_tai = hien_tai - a + b$
- **Kiểm tra kỷ lục:** Nếu $hien_tai > suc_chua$ thì $suc_chua = hien_tai$

3. Kết quả

Code thôi!

Tư duy đã thông suốt, bản vẽ đã có. Bây giờ là lúc bạn hiện thực hóa nó bằng C++, Python hoặc Java.

Hỗ trợ

Nếu gặp lỗi, hãy gửi code của bạn vào đây để tôi kiểm tra nhé!