

Coach Tư duy Thuật toán

Codeforces 1811A - Insert Digit

Slide Learning C++

Ngày 21 tháng 1 năm 2026

Lời mở đầu

Chào bạn!

Tôi đã nắm rõ vai trò là **Coach Tư duy Thuật toán** của bạn. Chúng ta sẽ cùng nhau "mổ xẻ" các bài toán theo phong cách **Micro-Chunks**, tập trung hoàn toàn vào tư duy logic và hình ảnh hóa vấn đề trước khi chạm vào code.

Hãy bắt đầu với bài **Codeforces 1811A - Insert Digit**.

Bước 1: Phân thuật đề bài (Deconstruct)

Loại bỏ những yếu tố rườm rà, đây là cốt lõi của vấn đề:

- **Dữ liệu cho sẵn:** Một số nguyên rất lớn (n chữ số) và một chữ số lẻ d (từ 0 đến 9).
- **Nhiệm vụ:** Chèn chữ số d vào **bất kỳ vị trí nào** để tạo ra một số mới **lớn nhất có thể**.
- **Lộ trình tư duy:**
 - 1 Hiểu quy tắc so sánh hai số lớn.
 - 2 Xác định thời điểm "vàng" để chèn chữ số d .
 - 3 Xử lý trường hợp chữ số d "về bết".

Bước 2: Mảnh ghép tư duy 1 - Quy tắc "Ai đứng trước là vua"

Ấn dụ

Những người đứng ở đầu hàng (bên trái) có quyền lực cao nhất. Bạn có một viên kim cương (chữ số d) và muốn đặt nó vào hàng sao cho tổng giá trị là lớn nhất.

Bẫy logic

Không phải lúc nào chèn vào cuối hay sau chữ số lớn nhất cũng là tối ưu. Ta cần so sánh trực tiếp d với từng "người" từ trái sang phải.

Thử thách tư duy 1

Giả sử số ban đầu là 764 và chữ số cần chèn là 5. Vị trí nào cho ra kết quả lớn nhất?

- A. Chèn vào đầu: 5764
- B. Chèn vào sau số 7: 7564
- C. Chèn vào sau số 6: 7654
- D. Chèn vào cuối: 7645

Thử thách tư duy 1

Giả sử số ban đầu là 764 và chữ số cần chèn là 5. Vị trí nào cho ra kết quả lớn nhất?

- A. Chèn vào đầu: 5764
- B. Chèn vào sau số 7: 7564
- C. Chèn vào sau số 6: 7654
- D. Chèn vào cuối: 7645

Đáp án: C. 7654

Ở hàng trăm, số 6 lớn hơn số 5. Vì vậy, 7645 và 7654 sẽ lớn hơn các số bắt đầu bằng 75...

Chiến thuật

Chúng ta sẽ duyệt từ trái sang phải, nếu thấy chữ số d của mình **lớn hơn** chữ số đang xét, thì đó chính là "thời điểm vàng" để chèn vào.

Thử thách tư duy 2: Nếu số 445 và $d = 6$. Bạn chèn vào đâu?

- A. Trước số 4 đầu tiên.
- B. Giữa hai số 4.
- C. Sau số 5.

Chiến thuật

Chúng ta sẽ duyệt từ trái sang phải, nếu thấy chữ số d của mình **lớn hơn** chữ số đang xét, thì đó chính là "thời điểm vàng" để chèn vào.

Thử thách tư duy 2: Nếu số 445 và $d = 6$. Bạn chèn vào đâu?

- A. Trước số 4 đầu tiên.
- B. Giữa hai số 4.
- C. Sau số 5.

Đáp án: A (Vì $6 > 4$, chèn ngay lập tức để có 6445).

Bước 3: Mảnh ghép tư duy 3 - Khi nào thì "Về bét"?

Trường hợp đặc biệt: Duyệt từ đầu đến cuối mà **không có chữ số nào nhỏ hơn d** .

Ví dụ: 887, $d = 3$.

- Duyệt 8 : $3 > 8$ (Sai)
- Duyệt 8 : $3 > 8$ (Sai)
- Duyệt 7 : $3 > 7$ (Sai)

Thử thách tư duy 3: Ta nên đặt số 3 ở đâu?

- A. Đặt ở ngay đầu số.
- B. Đặt ở cuối cùng của số.
- C. Không đặt vào nữa.

Bước 3: Mảnh ghép tư duy 3 - Khi nào thì "Về bết"?

Trường hợp đặc biệt: Duyệt từ đầu đến cuối mà **không có chữ số nào nhỏ hơn d** .

Ví dụ: 887, $d = 3$.

- Duyệt 8 : 3 > 8 (Sai)
- Duyệt 8 : 3 > 8 (Sai)
- Duyệt 7 : 3 > 7 (Sai)

Thử thách tư duy 3: Ta nên đặt số 3 ở đâu?

- A. Đặt ở ngay đầu số.
- B. Đặt ở cuối cùng của số.
- C. Không đặt vào nữa.

Đáp án: B (Kết quả: 8873).

Bước 4: Chốt thuật toán (The Master Plan)

- ➊ **Duyệt:** Từng chữ số của s từ trái sang phải.
- ➋ **So sánh:** Nếu chữ số $s[i] < d$, chèn d vào trước $s[i]$.
- ➌ **Dừng:** In phần còn lại và kết thúc.
- ➍ **Trường hợp "Về bết":** In d ở cuối cùng.

```
1 Doc vao so luong test case
2 Voi moi test case:
3   Doc n va d, xau s
4   Bien 'da_chen' = False
5   Duyet tung ky tu c trong chuoai s:
6     Neu 'da_chen' la False VA d > c:
7       In ra d
8       Bien 'da_chen' = True
9     In ra c
10  Neu 'da_chen' van la False:
11    In ra d
```

Listing 1: Ma gia thuan Viet

Thử thách cuối cùng

Số 543 và $d = 4$. Kết quả theo thuật toán là gì?

- A. 5443
- B. 4543
- C. 5434

Thử thách cuối cùng

Số 543 và $d = 4$. Kết quả theo thuật toán là gì?

- A. 5443
- B. 4543
- C. 5434

Giải thích

1. Xét số 5: $4 > 5$ (Sai).
2. Xét số 4: $4 > 4$ (Sai - lưu ý dấu $>$ chứ không phải \geq).
3. Xét số 3: $4 > 3$ (Đúng). Chèn ngay! → **5443**.

Bước 5: Hiện thực hóa ý tưởng (Implementation)

Vì số s có thể lên tới $2 \cdot 10^5$ chữ số, ta dùng **String**.

```
1 Nhập số lượng bộ test (t)
2 Lặp t lần:
3     Nhập n, d, xâu s
4     Tạo một biến có 'đã_chen' = false
5     Duyệt i từ 0 đến n-1:
6         Nếu 'đã_chen' == false VÀ d > (s[i] - '0'):
7             In ra d
8             'đã_chen' = true
9         In ra s[i]
10    Nếu 'đã_chen' vẫn là false:
11        In ra d
```

Câu hỏi chốt hạ: Tại sao cần $s[i] - '0'$?

- A. Biến ký tự (char) thành giá trị số để so sánh.
- B. Để xóa số 0 ở đầu.
- C. Quy tắc bắt buộc của Codeforces.

Bước 5: Hiện thực hóa ý tưởng (Implementation)

Vì số s có thể lên tới $2 \cdot 10^5$ chữ số, ta dùng **String**.

```
1 Nhap so luong bo test (t)
2 Lap t lan:
3     Nhap n, d, xau s
4     Tao mot bien co 'da_chen' = false
5     Duyet i tu 0 den n-1:
6         Neu 'da_chen' == false VA d > (s[i] - '0'):
7             In ra d
8             'da_chen' = true
9         In ra s[i]
10    Neu 'da_chen' van la false:
11        In ra d
```

Câu hỏi chốt hạ: Tại sao cần $s[i] - '0'$?

- A. Biến ký tự (char) thành giá trị số để so sánh.
- B. Để xóa số 0 ở đầu.
- C. Quy tắc bắt buộc của Codeforces.

Đáp án: A. Bạn có muốn tôi hướng dẫn code C++ hay Python không?