

Phân tích bài toán: 122A - Lucky Division

Tư duy thuật toán từng bước

Codeforces Solving

Ngày 20 tháng 1 năm 2026

Bước 1: Phẫu thuật đề bài (Briefing)

Các định nghĩa cốt lõi

- ➊ **Số may mắn (Lucky Number):** Là những số chỉ chứa các chữ số 4 và 7. Ví dụ: 47, 744, 4.
- ➋ **Số suýt may mắn (Almost Lucky Number):** Một số n được gọi là "suýt may mắn" nếu nó **chia hết** cho bất kỳ một "số may mắn" nào đó.
- ➌ **Nhiệm vụ:** Nhập n ($1 \leq n \leq 1000$). Kiểm tra xem n có phải là "số suýt may mắn" hay không.

Lộ trình tư duy

- **Chunk 1:** Nhận diện "Số may mắn" trong phạm vi cho phép.
- **Chunk 2:** Kiểm tra tính chia hết (Chiến thuật "Bắn hạ mục tiêu").
- **Chunk 3:** Tổng kết và chốt phương án.

Chunk 1: Nhận diện "Số may mắn"

Vì đề bài giới hạn n trong khoảng từ 1 đến 1000, chúng ta chỉ cần quan tâm các số may mắn trong khoảng này.

Liệt kê theo số lượng chữ số:

- **Có 1 chữ số:** 4, 7
- **Có 2 chữ số:** 44, 47, 74, 77
- **Có 3 chữ số:** 444, 447, 474, 477, 744, 747, 774, 777

Thử thách tư duy: Số nào KHÔNG phải "Lucky"?

Trong các số sau đây, số nào **KHÔNG PHẢI** là "số may mắn"?

- A. 47
- B. 744
- C. 467
- D. 7

Thử thách tư duy: Số nào KHÔNG phải "Lucky"?

Trong các số sau đây, số nào **KHÔNG PHẢI** là "số may mắn"?

- A. 47
- B. 744
- C. 467
- D. 7

Đáp án

C. 467.

Giải thích: Con số 6 đã "tố cáo" nó không phải là số may mắn (chỉ được chứa 4 và 7).

Chunk 2: Chiến thuật "Bắn hạ mục tiêu"

Để kiểm tra n có phải là "số suýt may mắn" hay không, ta xem n có chia hết cho **bất kỳ** số may mắn nào trong danh sách dưới đây không:

List = {4, 7, 44, 47, 74, 77, 444, 447, 474, 477, 744, 747, 774, 777}

Cạm bẫy (Trap)

Chỉ cần chia hết cho **MỘT** số trong danh sách trên là đủ để kết luận YES. Không cần phải chia hết cho tất cả!

Thử thách tư duy: Kiểm tra số 47

Giả sử người dùng nhập vào số **47**.

- 1 47 có chia hết cho số nào trong danh sách số may mắn không?
- 2 Kết quả cuối cùng bài này nên in ra là YES hay NO?

Thử thách tư duy: Kiểm tra số 47

Giả sử người dùng nhập vào số **47**.

- 1 47 có chia hết cho số nào trong danh sách số may mắn không?
- 2 Kết quả cuối cùng bài này nên in ra là YES hay NO?

Kết quả

YES.

Giải thích: 47 chia hết cho 47 (chính nó là một số may mắn). Theo định nghĩa, 47 là "số suýt may mắn".

Chunk 3: Kiểm tra số 79

Nếu nhập vào số **79**. Hãy nhìn lại danh sách:
4, 7, 44, 47, 74, 77, ...

Theo bạn, với $n = 79$, máy tính sẽ in ra YES hay NO?

Chunk 3: Kiểm tra số 79

Nếu nhập vào số **79**. Hãy nhìn lại danh sách:
4, 7, 44, 47, 74, 77, ...

Theo bạn, với $n = 79$, máy tính sẽ in ra YES hay NO?

Kết quả

NO.

Giải thích: 79 không chia hết cho 4, không chia hết cho 7, và cũng không chia hết cho bất kỳ số nào lớn hơn trong danh sách.

Mã giả (Tư duy lập trình)

Chiến thuật "Duyệt danh sách" được thể hiện như sau (Tiếng Việt không dấu):

```
1 BUOC 1: Tao mot mang (danh sach) chua cac so:
2         4, 7, 44, 47, 74, 77, 444, 447, 474, 477, 744, 747...
3
4 BUOC 2: Nhap so nguyen n tu ban phim.
5
6 BUOC 3: Tao bien "kiem_tra" va dat la False (chua tim thay).
7
8 BUOC 4: Chay vong lap qua tung so "x" trong danh sach o Buoc 1:
9         Neu n chia cho x du 0:
10            Dat "kiem_tra" thanh True.
11            Dung vong lap (break).
12
13 BUOC 5: Neu "kiem_tra" la True:
14         In ra "YES".
15         Nguoc lai:
16         In ra "NO".
```

Listing 1: Mã giả giải thuật

Câu hỏi chốt thuật toán

Trong Bước 4, tại sao chúng ta nên dùng lệnh "Dừng vòng lặp"(break) ngay khi tìm thấy một số mà n chia hết?

- 1 Để tiết kiệm thời gian chạy của máy tính.
- 2 Vì nếu không dừng, máy tính sẽ bị lỗi.
- 3 Vì đề bài yêu cầu chỉ được chia hết cho đúng một số duy nhất.

Câu hỏi chốt thuật toán

Trong Bước 4, tại sao chúng ta nên dùng lệnh "Dừng vòng lặp"(break) ngay khi tìm thấy một số mà n chia hết?

- 1 Để tiết kiệm thời gian chạy của máy tính.
- 2 Vì nếu không dừng, máy tính sẽ bị lỗi.
- 3 Vì đề bài yêu cầu chỉ được chia hết cho đúng một số duy nhất.

Đáp án chính xác

1. Để tiết kiệm thời gian chạy của máy tính.

Khi đã tìm thấy một ước số may mắn, điều kiện đề bài đã thỏa mãn. Việc kiểm tra tiếp là dư thừa.