

Huấn luyện viên Tư duy Thuật toán

Codeforces 1472B - Fair Division

Gemini AI

2026

Phẫu thuật đề bài (Briefing)

Cốt lõi vấn đề

- **Dữ liệu:** Túi kẹo gồm 2 loại: nặng 1g và nặng 2g.
- **Mục tiêu:** Chia kẹo thành 2 phần có **tổng khối lượng** bằng nhau.
- **Ràng buộc:** Không được cắt đôi viên kẹo.

Lộ trình tư duy

- ❶ Điều kiện cần (Tổng khối lượng).
- ❷ Khả năng lấp đầy (Sử dụng kẹo 2g và kẹo 1g).
- ❸ Xử lý bẫy logic.

Chunk 1: Điều kiện "Tổng chẵn"

- Để chia đôi thành 2 phần bằng nhau (A và B), thì mỗi phần phải nặng đúng bằng $\frac{S}{2}$.
- Ảnh dụ:** Nếu có tổng 7 viên kẹo 1g, không thể chia đều mà không bẻ đôi.

Thử thách tư duy

Túi có: 1 viên 2g và 1 viên 1g. Tổng khối lượng là 3g. Có thể chia đều không?

Chunk 1: Điều kiện "Tổng chẵn"

- Để chia đôi thành 2 phần bằng nhau (A và B), thì mỗi phần phải nặng đúng bằng $\frac{S}{2}$.
- Ảnh dụ:** Nếu có tổng 7 viên kẹo 1g, không thể chia đều mà không bẻ đôi.

Thử thách tư duy

Túi có: 1 viên 2g và 1 viên 1g. Tổng khối lượng là 3g. Có thể chia đều không?

Đáp án: Không! Vì 3 là số lẻ.

Điều kiện 1

Tổng khối lượng tất cả viên kẹo phải là một **số chẵn**.

Chunk 2: "Bấy" kẹo 2g

Trường hợp ngang trái

Bạn có 1 viên kẹo 2g và 0 viên kẹo 1g. Tổng là 2 (số chẵn). Có chia được không?

Chunk 2: "Bẫy" kẹo 2g

Trường hợp ngang trái

Bạn có 1 viên kẹo 2g và 0 viên kẹo 1g. Tổng là 2 (số chẵn). Có chia được không?

Đáp án: Không! Vì chỉ có 1 khối 2g, không thể chia cho 2 người mỗi người 1g.

Quan sát quan trọng

- Nếu số lượng kẹo 2g là **chẵn**: Chia đều dễ dàng.
- Nếu số lượng kẹo 2g là **lẻ**: Dư ra một viên 2g. Cần kẹo 1g để bù đắp.

Chunk 2: "Bấy" kẹo 2g

Trường hợp ngang trái

Bạn có 1 viên kẹo 2g và 0 viên kẹo 1g. Tổng là 2 (số chẵn). Có chia được không?

Đáp án: Không! Vì chỉ có 1 khối 2g, không thể chia cho 2 người mỗi người 1g.

Quan sát quan trọng

- Nếu số lượng kẹo 2g là **chẵn**: Chia đều dễ dàng.
- Nếu số lượng kẹo 2g là **lẻ**: Dư ra một viên 2g. Cần kẹo 1g để bù đắp.

Câu hỏi

Cần ít nhất bao nhiêu viên 1g để bù cho 1 viên 2g dư?

Chunk 2: "Bấy" kẹo 2g

Trường hợp ngang trái

Bạn có 1 viên kẹo 2g và 0 viên kẹo 1g. Tổng là 2 (số chẵn). Có chia được không?

Đáp án: Không! Vì chỉ có 1 khối 2g, không thể chia cho 2 người mỗi người 1g.

Quan sát quan trọng

- Nếu số lượng kẹo 2g là **chẵn**: Chia đều dễ dàng.
- Nếu số lượng kẹo 2g là **lẻ**: Dư ra một viên 2g. Cần kẹo 1g để bù đắp.

Câu hỏi

Cần ít nhất bao nhiêu viên 1g để bù cho 1 viên 2g dư?

Chính xác: Cần ít nhất **2 viên 1g**.

Chunk 3: Tổng kết logic

Gọi n_1 là số lượng kẹo 1g, n_2 là số lượng kẹo 2g.

- **Trường hợp A:** Nếu n_2 chẵn. Chỉ cần n_1 chẵn là xong.
- **Trường hợp B:** Nếu n_2 lẻ. Cần ít nhất 2 viên 1g để cân bằng, sau đó số kẹo 1g còn lại ($n_1 - 2$) cũng phải chẵn.

Thử thách chốt hạ

$n_2 = 3$ và $n_1 = 1$. Tổng = 7g. Có chia được không?

Chunk 3: Tổng kết logic

Gọi n_1 là số lượng kẹo 1g, n_2 là số lượng kẹo 2g.

- **Trường hợp A:** Nếu n_2 chẵn. Chỉ cần n_1 chẵn là xong.
- **Trường hợp B:** Nếu n_2 lẻ. Cần ít nhất 2 viên 1g để cân bằng, sau đó số kẹo 1g còn lại ($n_1 - 2$) cũng phải chẵn.

Thử thách chốt hạ

$n_2 = 3$ và $n_1 = 1$. Tổng = 7g. Có chia được không?

Đáp án: Không! Vì tổng khối lượng lẻ.

Mã giả (Pseudocode)

```
1     Neu (Tong_khoi_luong la so le):
2         In ra "NO"
3     Neu (So_luong_keo_2g la so le VA So_luong_keo_1g == 0):
4         In ra "NO"
5     Con lai:
6         In ra "YES"
7
```

Listing 1: Logic kiểm tra điều kiện

Giải thích

Nếu tổng chẵn và có đủ kẹo 1g để bù đắp cho sự lẻ loi của kẹo 2g (nếu có), chúng ta luôn chia được.

Mở rộng: Tư duy Quy hoạch động (DP)

Bản chất DP

Sử dụng mảng $dp[i]$ để đánh dấu xem có thể tạo ra tổng khối lượng bằng i hay không.

Đặc điểm	Greedy/Math	Quy hoạch động (DP)
Tốc độ	Cực nhanh $O(1)$	Chậm hơn $O(n \cdot \text{target})$
Độ linh hoạt	Thấp	Rất cao (nhiều loại tạ)

Cấu trúc mã giả DP

```
1     target = tong_khoi_luong / 2
2     dp[0] = True
3     Cac dp[1...target] = False
4
5     Voi moi vien keo v trong danh sach:
6         Chay i tu target xuong den v:
7             dp[i] = dp[i] HOAC dp[i - v]
8
9     Ket qua: Neu dp[target] la True thi YES, nguoc lai NO
10
```

Listing 2: Thuật toán Dynamic Programming

Câu hỏi tư duy

Tại sao biến i phải chạy **ngược** từ target về v ?

Cấu trúc mã giả DP

```
1     target = tong_khoi_luong / 2
2     dp[0] = True
3     Cac dp[1...target] = False
4
5     Voi moi vien keo v trong danh sach:
6         Chay i tu target xuong den v:
7             dp[i] = dp[i] HOAC dp[i - v]
8
9     Ket qua: Neu dp[target] la True thi YES, nguoc lai NO
10
```

Listing 3: Thuật toán Dynamic Programming

Câu hỏi tư duy

Tại sao biến i phải chạy **ngược** từ target về v ?

Gợi ý: Để đảm bảo mỗi viên kẹo chỉ được sử dụng **một lần** duy nhất.