Lời giải đề thi thử lần 2 (Tác giả: Tạ Chí Thành Danh)

Bài 1 – CATTHEP

Tóm tắt đề: Cho một số nguyên dương L với ($12 \le L \le 200$), hãy đếm số cách biểu diễn L dưới dạng tổng của 12 số nguyên dương.

Lưu ý: Hai cách biểu diễn của nó được gọi là khác nhau nếu tại tại vị trí thứ i nào đó trong hai cách biểu diễn (tạm gọi là A, B) thì A_i khác B_i.

Ví dụ như đối với số 13 thì hai cách này là khác nhau:
1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 2

2 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1

Lời giải:

Để thuận tiện cho việc giải thích, số L anh sẽ ký hiệu là N và số các số nguyên dương trong một cách biểu diễn là K = 12.

Cách 1: Sử dụng công thức toán tổ hợp

Đây là một trường hợp đặc biệt của bài toán sau đây (từng được đem ra bàn luận trên diễn đàn Brainly và Quora), với lưu ý là N và K ở trong bài này khá nhỏ (N tối đa 200 và K = 12). Vì thế, thay vì chúng ta phải dùng quy hoạch động để tính tổ hợp chập K-1 của N-1 phần tử (ký hiệu $\binom{N-1}{K-1}$) thì ta có thể tính nhanh bằng công thức sau đây:

b) Số các tổ hợp

Kí hiệu C_n^k (hoặc $\binom{n}{k}$) là số các tổ hợp chập *k* của một tập hợp có *n* phần tử.

ĐỊNH LÍ 3

Số các tổ hợp chập *k* của một tập hợp có *n* phần tử ($1 \le k \le n$) là

$$C_n^k = \frac{A_n^k}{k!} = \frac{n(n-1)(n-2)\ldots(n-k+1)}{k!}. \qquad (3)$$

Với K! = K(K-1)(K-2)…3.2.1

Độ phức tạp tính toán: O(K)

Lưu ý: Vì bài này giới hạn khá nhỏ ($12 \le N \le 200$ và K = 12) nên đáp án sẽ không vượt quá giới hạn của số nguyên 64 bit, tuy nhiên, trong một số trường hợp, N và K đủ lớn sẽ khiến cho đáp án sẽ rất lớn để sẽ yêu cầu ta in kết quả là **phần dư** với **10^9+7 (hoặc 998244353)**, đồng nghĩa với việc ta phải kết hợp **kiến thức nghịch đảo modulo, hàm**

lũy thừa nhanh và số học đồng dư (modular arithmetic) để tính toán ra kết quả chính xác, và hai chủ đề này khá là khó đối với trình độ của Olympic không chuyên, nên anh không đánh giá cao cách giải này cho lắm (~~chắc chắn không phải vì anh ngu toán tổ hợp và lý thuyết số đâu~~ 😊).

Link code mẫu

Cách 2: Sử dụng thuật toán Quy hoạch động

Dễ thấy bài toán này có thể đưa về bài toán quy hoạch động cái túi kinh điển, và thậm chí đã có một trang web đề cập đến hướng giải bằng quy hoạch động cho bài này. Tuy nhiên, cách làm của anh sẽ khác một chút và anh xin trình bày ngay sau đây.

Gọi DP[K][N] là số cách để tạo thành tổng N với K số nguyên dương, ta có công thức truy hồi như sau:

$$\mathbf{DP[K][N]} = \sum_{i=1}^{N} \mathbf{DP[K-1][N-i]}$$

Giải thích: Số cách hình thành nên tổng N với K số nguyên dương sẽ bằng số cách hình thành tổng N – i với K – 1 số nguyên dương trước đó, với i nằm trong khoảng từ 1 đến N (số nguyên dương thành phần không vượt quá N). Ở đây thì N tối đa là 200 và K = 12 nên ta chỉ cần tạo mảng DP[13][201] (0-based) để xử lí trước phần tính toán là được.

Độ phức tạp tính toán: $O(K \cdot N^2)$

Lưu ý: Bài này có thể tối ưu xuống còn O(K·N), nhưng không cần thiết vì giới hạn rất nhỏ.

Link code mẫu

Update (Thứ năm 25/03/2021): Vì hôm nay anh khá rảnh nên anh sẽ viết thêm solution O(K·N).

Như mọi người thấy trong thuật $O(K \cdot N^2)$, thuật toán trên bao gồm hai bước:

1. Chạy 2 vòng lặp lồng nhau ở ngoài để duyệt qua từng trường hợp của DP[K][N] trong O(K·N).

2. Tính DP[K][N] theo như công thức truy hồi chạy trong khoảng O(N) vì duyệt từ i đến N để cộng DP[K – 1][N – i] vào mảng.

Xem lại 2 bước trên, bước 1 rất khó để tối ưu xuống, do đó ta nên tập trung vào bước 2 để giảm độ phức tạp thuật toán. Nhìn kỹ lại công thức QHĐ thì ta thấy một điều rằng, việc tính DP[K][N] ứng với mỗi K và N xác định đều phải lặp liên tục i từ 1 đến N để tính tổng và thêm tổng số cách đó vào DP[K][N], nếu để ý kỹ thì ở đây ta có thể tăng tốc thuật bằng cách lưu giá trị của những trạng thái trước vào trạng thái hiện tại trong một lần tính bằng cách áp dụng mảng cộng dồn (prefix sum) thay vì phải duyệt lại cả mảng từ i = 1 đến vị trí N (vị trí hiện tại). Vì thế công thức truy hồi của chúng ta trong mỗi lần tính sẽ như sau:

DP[K][N] = DP[K – 1][N – 1] + DP[K][N – 1]

Giải thích: Số cách hình thành nên tổng N với K số nguyên dương lúc này sẽ bằng số cách DP[K – 1][N - 1] (lúc này không cần phải duyệt từ 0 đến N - 1 vì sử dụng thêm công thức mảng cộng dồn để lưu các giá trị trước đó vào giá trị gần nhất) cộng cho DP[K][N – 1] (số cách tạo nên tổng N – 1 với K số nguyên dương). Nếu vẫn không hiểu thì nên ngẫm thêm một tí và xem thêm ở đây.

Link code mẫu

Bài 2 – NYT

Nguồn bài: Codeforces Good Bye 2014

Lời giải:

- Official Editorial #1
- Official Editorial #2
- Lời giải chi tiết bằng tiếng Việt

Code mẫu (đã bao gồm ở trong link đầu tiên).

Bên dưới là code của anh.

DFS (tachithanhdanh)

Non-DFS (tachithanhdanh)

Editorial này được viết vào **Chiều thứ Hai ngày 22/03/2021.**

Update 1: Chỉnh sửa lại (thêm solution DP-Optimised cho bài 1) vào **Tối thứ Năm ngày 25/03/2021.**