Bitmasking and Dynamic Programming | Set 1 (Đếm số cách gán một chiếc mũ duy nhất cho mỗi người)

Hãy xem xét các vấn đề được nêu dưới đây:

Có 100 loại mũ khác nhau, mỗi loại có một số ID duy nhất được đánh số từ 1 đến 100. Ngoài ra, có N người và mỗi người có một bộ suy tập với số lưỡng mũ khác nhau. Một ngày nọ, tất cả những người này quyết định tham gia một bữa tiệc đội mũ nhưng để cho độc đáo, họ quyết định rằng không ai trong số họ sẽ đội cùng một loại mũ. Vì vậy, hãy đếm tổng số cách sắp xếp sao cho không có bất kì ai trong số họ mang cùng một loại mũ.

Ràng buộc: 1 <= N <= 10. Ví dụ:



Bởi vì đầu ra rất có thể là rất lớn nên modulo cho 1e9 + 7.

Một giải pháp đơn giản là cố gắng thử tất cả các trường hợp có thể. Bắt đầu bằng cách chọn phần tử đầu tiên từ bộ đầu tiên, đánh dấu nó là đã truy cập và tiến hành lặp lại cho các bộ còn lại. Về cơ bản đây là một giải pháp dựa trên Bachtracking.

Một giải pháp tốt hơn là sử dụng Bitmasking and DP. Trước tiên chúng ta hãy tìm hiểu về Bitmasking.

Bitmasking là gì?

Giả sử chúng ta có một tập hợp các phần tử được đánh số từ 1 đến N. Nếu chúng ta muốn biểu diễn một tập hợp con của tập hợp này thì nó có thể được mã hóa bằng một chuổi bao gồm N bit (chúng ta thường hay gọi chuổi này là “mask”). Trong tập hợp con đã chọn của chúng ta, phần thử thứ i-th thuộc về nó khi và chỉ khi bit thứ i-th của mask này bằng 1. Ví dụ, chúng ta có mask 10000101, thì tập hợp con của tập hợp [1, 2, …, 8] bao gồm các phần tử 1, 3 và 8. Chúng ta biết rằng, một tập hợp gồm N phần tử thì sẽ có tổng cộng tập hợp con do đó việc có cái mask là có thể, một đại diện cho mỗi tập hơp con. Trong thực tế, mỗi mask là một số nguyên được viết bằng kí hiệu nhi phân.

Phương pháp chính của chúng ta ở đây là gán một giá trị cho mỗi mask (cho mỗi một tập hợp con) và như thế thực hiện tính toán các giá trị cho các mask mới bằng cách sử dụng lại các giá trị cuả các mask đã được tính toán. Thông thường mục tiêu chính của chúng ta là tính toán giá trị/giải pháp cho tập hợp hoàn chỉnh, tức là cho mask 11111111. Thông thường để tìm giá trị cho một tập hợp con X, chúng ta sẽ loại bỏ một phần tử bằng mọi cách có thể và sử dụng những giá trị này cho các tập hợp con thu được để tính toán giá trị/giải pháp cho . Điều này có nghĩa rằng giá trị phải được tính toán rồi. Vậy nên chúng ta cần thiết lập một trật tự mà các mask sẽ được xem xét. Thật dễ dàng để hiểu với cách sắp xếp tự nhiên sẽ làm: đi qua các mask theo thứ tự tăng dần của các số tương ứng. Ngoài ra, chúng ta cũng có thể bắt đầu với một tập hợp con rỗng và chúng ta sẽ thêm các phần tử theo mọi cách có thể và sử dụng các giá trị của các tập hơp con đã thu được để tính toán giá trị/giải pháp cho .

Chúng tôi sẽ chủ yếu sử dụng các kí hiệu/thao tác trên mask như sau:

bit(i, mask) – bit thứ i-th của mask

count(mask) – số lượng bit khác 0 trong mask

first(mask) – số của bit khác 0 thấp nhất trong mask

set(i, mask) – thiết lập bit thứ i-th trong mask

check(i, mask) – kiểm tra bit thứ i-th trong mask

Vấn đề này sẽ được giải quyết bằng Bitmasking + DP như thế nào?

Ý tưởng sử dụng từ thực tế là có tới 10 người. Vì vậy, chúng ta có thể sử dụng một biến số nguyên như một bitmask để lưu trữ người nào đang đội mũ và người nào không.