



George and Round

Link submit: <http://codeforces.com/problemset/problem/387/B>

Solution:

C++	http://ideone.com/LQIDK5
Java	https://ideone.com/nVUgCe
Python	https://ideone.com/37iKjU

Tóm tắt đề:

George quyết định chuẩn bị một kỳ thi cho Codesecrof, cho nên anh ấy đã chuẩn bị m problem cho kỳ thi.

Để có được một kỳ thi hay, thì cần phải có ít nhất n problem, đồng thời có ít nhất một problem có độ phức tạp a_1 , ít nhất một problem có độ phức tạp a_2 , ..., ít nhất một problem có độ phức tạp a_n và các problem có độ phức tạp khác nhau.

George có thể đơn giản hóa bất kỳ problem nào đã chuẩn bị với độ phức tạp c thành độ phức tạp d ($c \geq d$) bằng cách thay đổi giới hạn dữ liệu đầu vào.

Cho số lượng problem hiện tại George đang có, bạn cần kiểm tra xem với những problem mà George đang có thì cần thêm bao nhiêu problem nữa để tạo ra được một kỳ thi tốt.

Mỗi problem được đại diện là một con số, số càng lớn thì độ phức tạp càng lớn.

Input:

Dòng đầu tiên chứa 2 số n và m ($1 \leq n, m \leq 3000$) – n là cho số problem tối thiểu cần có trong kỳ thi, m là số problem hiện có.

Dòng thứ hai chứa n số nguyên ($1 \leq a_1 < a_2 < \dots < a_n \leq 10^6$) ứng với độ phức tạp cần có của các problem trong kỳ thi.

Dòng thứ ba chứa m số nguyên ($1 \leq b_1 \leq b_2 \leq \dots \leq b_m \leq 10^6$) ứng với độ phức tạp của các problem hiện có.

Output:

In ra một số duy nhất là số lượng problem cần chuẩn bị thêm cho kỳ thi nếu George không chuẩn bị đủ.

Ví dụ:

3 5 1 2 3 1 2 2 3 3	0
---------------------------	---

Giải thích ví dụ:

Kỳ thi cần có 3 bài với cấp độ 1, 2, 3. George đã chuẩn bị 5 bài. Trong 5 bài của George đã có đủ các bài với độ phức tạp trên nên không cần thêm bài nào. Xuất ra giá trị 0.

Hướng dẫn giải:

Lần lượt bỏ độ khó của các bài vào vector a (bài toán sẽ ra) và vector b là các bài toán có sẵn.

Dùng 2 vòng lặp, vòng lặp đầu tiên sẽ chạy qua lần lượt các độ khó của vector a, trong lúc chạy xét xem phần tử nào trong b có thể dùng để ra đề thì tăng biến đếm lên.

Kết quả (n – biến đếm) chính là số bài cần chuẩn bị thêm.

Độ phức tạp: $O(\max(n,m))$ với n, m lần lượt là số problem kỳ thi cần và số problem hiện có.