



A. Array

Link submit: <http://codeforces.com/problemset/problem/572/A>

Solution:

C++: <http://ideone.com/cU8NXR>

Java: <https://ideone.com/99cRJ3>

Python: <https://ideone.com/lgcDAI>

Tóm tắt đề: Cho 2 mảng số nguyên A và B được sắp xếp không giảm. Kiểm tra xem liệu **có thể** chọn k số lượng phần tử trong mảng A và m số lượng phần tử trong mảng B sao cho bất kỳ phần tử nào trong mảng A cũng nhỏ hơn một phần tử bất kỳ được chọn trong mảng B.

Input:

Dòng đầu tiên chứa 2 số n_A, n_B ($1 \leq n_A, n_B \leq 10^5$), là số lượng phần tử của mảng A và mảng B.

Dòng tiếp theo chứa 2 số k và m ($1 \leq k \leq n_A, 1 \leq m \leq n_B$)

Dòng tiếp theo nữa lần lượt là các phần tử của mảng A. ($-10^9 \leq a_1 \leq a_2 \leq \dots \leq a_{n_A} \leq 10^9$),

Dòng cuối cùng là các phần tử của mảng B. ($-10^9 \leq b_1 \leq b_2 \leq \dots \leq b_{n_B} \leq 10^9$)

3 3	YES
2 1	
1 2 3	
3 4 5	

Giải thích: Mảng A có 3 phần tử và mảng B cũng vậy. Có thể lấy 2 phần tử ở mảng A và 1 phần tử ở mảng B sao cho 2 phần tử ở mảng A luôn luôn nhỏ hơn phần tử mảng B được không?

Hướng dẫn giải:

Trong các phần tử của mảng A chọn k phần tử nhỏ nhất. Sau đó lấy phần tử lớn nhất trong k phần tử này $a[k-1]$.

Trong các phần tử của mảng B chọn m phần tử lớn nhất. Sau đó lấy phần tử nhỏ nhất trong m phần tử này ra $b[nb-m]$.

Nếu Phần tử nhỏ nhất của B đã lớn hơn phần tử lớn nhất của A thì mọi phần tử trong k phần tử của mảng A luôn luôn nhỏ hơn m phần tử trong mảng B. Kết quả là YES.

Ngược lại thì kết quả là NO.

Công thức: $a[k-1] < b[nb-m]$

Độ phức tạp: $O(1)$

Big-O Coding