Monk and his Friends

Link: https://www.hackerearth.com/practice/data-structures/trees/binary-search-tree/practice-problems/algorithm/monk-and-his-friends/description/

Solution:

C++	http://ideone.com/PZm9lR
Java	https://ideone.com/ZJbIx0
Python	https://ideone.com/6qZFak

Tóm tắt đề: Có N sinh viên trong lớp học. Có M sinh viên khác đang đến, khi mỗi học sinh này đến, họ sẽ tìm đến ngồi cạnh với các học sinh khác đã có trong lớp mà số kẹo của họ đúng bằng nhau. Với mỗi học sinh trong M học sinh đó, in ra YES nếu như người đó có thể tìm được chỗ ngồi cạnh người có cùng số kẹo với mình, ngược lại thì in ra NO.

Input

Dòng đầu tiên chứa T ($1 \le T \le 10$)số lượng bộ test, mỗi bộ test gồm các thông tin sau:

Dòng thứ nhất chứa 2 số cách nhau bởi dấu khoảng trắng là N và M ($1 \le N, M \le 10^5$)

Dòng thứ hai chứa N + M số lần lượt là N sinh viên và M sinh viên khác ($0 \le Ai \le 10^{12}$)

Output

Mỗi bộ test bạn sẽ in ra M dòng mới, trả lời cho M sinh viên.

In ra YES nếu tìm thấy và NO nếu không tìm thấy.

1 2 3 3 2 9 11 2	NO NO YES

Giải thích:

Có 2 sinh viên đã có sẵn trong lớp (3, 2).

- Sinh viên 9 đến: NO.
- Sinh viên 11 đến: NO.
- Sinh viên 2 đến: YES.

Hướng dẫn giải:

- Đầu tiên bạn sẽ bỏ N giá trị sinh viên vào trong set.
- Sau đó bạn lần lượt bỏ các sinh viên trong M sinh viên vào trong set. Nếu đã có rồi thì bạn xuất ra YES. Nếu chưa thì xuất ra NO.

Độ phức tạp: O(T * M * log(N + M)) với N và M lần lượt là số lượng sinh viên đã có sẵn và số lượng sinh viên vào lớp học và T là số lượng bộ test.

Lưu ý: Đối với C++ thì bài này yêu cầu xuất dữ liệu rất nhiều, nên để có thể Accepted được thì bạn cần giảm chi phí thời gian khi xuất dữ liệu, có 2 cách là sử dụng "\n" thay vì dùng endl hoặc thêm dòng lệnh: ios_base::sync_with_stdio(false); ở đầu chương trình.

"\n" khác endl ở chỗ endl ngoài việc xuống dòng, nó còn đưa toàn bộ dữ liệu trong buffer output stream ra output stream (tức xuất kết quả) mặc dù buffer (bộ nhớ đệm) vẫn còn trống, việc cứ phải đưa vào bộ nhớ đệm và lấy ra, xóa nhiều lần như vậy ảnh hưởng đến tốc độ xử lý. Còn "\n" chỉ xuống dòng, khi nào buffer đầy thì mới đưa ra output stream, giảm thiểu được số lần giải phóng bộ nhớ đệm.

Việc thêm dòng lệnh trên dùng để tăng tốc cho cin, cout của C++ nhanh như C bình thường vì bản chất cin, cout chậm hơn scanf và printf rất nhiều. Bạn có thể đọc thêm tài liệu ở đây.

https://www.quora.com/What-is-use-of-the-statement-ios_base-sync_with_stdio-false-cin-tie-NULL-cout-tie-NULL-What-does-it-do