



## B. Big Segment

Link submit: <http://codeforces.com/problemset/problem/242/B>

Solution:

C++	<a href="http://ideone.com/BtmKHy">http://ideone.com/BtmKHy</a>
Java	<a href="https://ideone.com/Sorvl8">https://ideone.com/Sorvl8</a>
Python	<a href="https://ideone.com/5xISme">https://ideone.com/5xISme</a>

**Tóm tắt đề:** Cho bạn  $n$  đoạn thẳng ( $1 \leq n \leq 10^5$ ) Nhiệm vụ của bạn là tìm ra **một đoạn** mà nó có thể phủ hết các đoạn còn lại. Giá trị của mỗi đoạn được đại diện là Left và Right  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq 10^9$ ). Nếu không có đoạn nào có thể phủ các đoạn còn lại thì xuất ra giá trị -1.

**Input:**

Dòng đầu tiên chứa số  $n$  là số lượng đoạn bạn có.

Các dòng tiếp theo lần lượt chứa  $n$  cặp số là các đoạn thẳng. Mỗi cặp số nằm trên 1 dòng

**Output:**

In ra đoạn có thể phủ hết các đoạn còn lại ngược lại in ra -1.

6 1 5 2 3 1 10 7 10 7 7 10 10	3
---	---



**Giải thích:** Có 6 đoạn trong đó có đoạn ở vị trí số 3 là  $[1, 10]$  là có thể chứa toàn bộ các đoạn còn lại. Vì thế in ra giá trị 3.

### Hướng dẫn:

Bài này sử dụng 2 dòng for.

- Dòng for đầu tiên bạn sẽ tìm giá trị nhỏ nhất và lớn nhất của Left và Right theo logic: Left sẽ là số nhỏ nhất trong mảng L và Right sẽ là số lớn nhất trong mảng R.
- Dòng for thứ 2 bạn sẽ tìm trong các đoạn, đoạn nào trùng với Left và Right bạn vừa tìm ở trên, in ra kết quả và số thứ tự của đoạn đó. Ngược lại nếu không có đoạn nào trùng với Left và Right nghĩa là không tồn tại đoạn nào phủ hết các đoạn còn lại.

**Độ phức tạp:**  $O(n)$  với  $n$  là số lượng đoạn thẳng.

Big-O Coding