You are implementing a program to use as your calendar. We can add a new event if adding the event will not cause a **double booking**.

A **double booking** happens when two events have some non-empty intersection (i.e., some moment is common to both events.).

The event can be represented as a pair of integers start and end that represents a booking on the half-open interval [start, end), the range of real numbers x such that start <= x < end.

Implement the MyCalendar class:

* MyCalendar() Initializes the calendar object.
* boolean book(int start, int end) Returns true if the event can be added to the calendar successfully without causing a **double booking**. Otherwise, return false and do not add the event to the calendar.

**Example 1:**

Input  
["MyCalendar", "book", "book", "book"]  
[[], [10, 20], [15, 25], [20, 30]]  
Output  
[null, true, false, true]  
  
Explanation  
MyCalendar myCalendar = new MyCalendar();  
myCalendar.book(10, 20); // return True  
myCalendar.book(15, 25); // return False, It can not be booked because time 15 is already booked by another event.  
myCalendar.book(20, 30); // return True, The event can be booked, as the first event takes every time less than 20, but not including 20.

**Constraints:**

* 0 <= start < end <= 109
* At most 1000 calls will be made to book.