Smallest Bookcase (Easy)

Problem category:	Sorting, Binary Search, Dynamic Programming
Expected difficulty:	2000

Solution

From the problem statement, we know that two books will be in conflict when one book is wider and shorter than the other, or vice-versa. So, basically, we want to find the largest set of books that are mutually in conflict (i.e., no two books from this set can be on the same shelf). The solution to this problem is known as the longest increasing subsequence (LIS).

There are various ways to implement it. Here's one possible algorithm:

1. Sort the books by width, then by height. Now we can safely ignore the width. Notice that the heights form a sawtooth pattern.
2. Take each book from this series, in order, and add it to the shelf whose last added book is closest in height to, but no larger than, the current one.
3. Once the current book height becomes smaller than the smallest among all shelves so far, we need to create another shelf.

Complexity

Since we need to sort the input, as well as perform a binary search for each book, the overall time complexity is $O(n \log n)$. The memory complexity is $O(n)$.