# Small Bookcase

From the statement, we know that two books are in conflict if one book is wider and shorter than the other. Essentially, we want to find the largest set of books that are mutually in conflict: i.e., a set from which no pair of books can appear on the same shelf.

This problem can be reduced to the longest increasing subsequence (LIS), for which various solutions exist. Here's one possible algorithm:

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

Since the input must first be sorted, and because each book requires a binary search, the overall time complexity is $O(n \log n)$.