

书籍叠放题目描述：

书籍的长、宽都是整数对应  $(l,w)$ 。如果书  $A$  的长宽度都比  $B$  长宽大时，则允许将  $B$  排列放在  $A$  上面。

现在有一组规格的书籍，书籍叠放时要求书籍不能做旋转，请计算最多能有多少个规格书籍能叠放在一起。

输入描述：

输入:  $books = [[20,16],[15,11],[10,10],[9,10]]$

说明：总共 4 本书籍，第一本长度为 20 宽度为 16；第二本书长度为 15 宽度为 11，依次类推，最后一本书长度为 9 宽度为 10。

输出描述：

输出: 3

说明：最多 3 个规格的书籍可以叠放到一起，

从下到上依次为:  $[20,16],[15,11],[10,10]$

补充说明：

```
import java.util.*;
public class Main {

    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        String tmp = sc.nextLine();
        String s ="";
        for (int i =0;i<tmp.length(); ++i){
            if (tmp.charAt(i) != ' '){
                s += tmp.charAt(i);
            }
        }
        String [] x = s.substring(1,s.length()-1).split(",");

        int n =x.length;
        int [][] arr = new int[n/2][2];
        for(int i = 0;i < n;i += 2){
```

```

        arr[i / 2][0] = Integer.parseInt(x[i].substring(1));
        arr[i / 2][1] = Integer.parseInt(x[i + 1].substring(0, x[i + 1].length() - 1));
    }
    Arrays.sort(arr, (a, b) -> {
        if (a[0] != b[0])
            return a[0] - b[0];
        return b[1] - a[1];
    });

    ArrayList<Integer> dp = new ArrayList<>();
    dp.add(arr[0][1]);
    for (int i = 1; i < arr.length; ++i) {
        int y = arr[i][1];
        if (y > dp.get(dp.size() - 1)) {
            dp.add(y);
            continue;
        }
        int left = 0, right = dp.size() - 1;
        while (left < right) {
            int mid = (left + right) >> 1;
            if (dp.get(mid) < y) {
                left = mid + 1;
            } else {
                right = mid;
            }
        }
        dp.set(left, y);
    }
    System.out.println(dp.size());
}
}

```