

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: $[20,16],[15,11],[10,10]$

示例 1

输入：

$[[20,16],[15,11],[10,10],[9,10]]$

输出：

3

说明：

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: $[20,16],[15,11],[10,10]$

```
import java.util.*;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        // [[0,0]]
```

```
        Scanner in = new Scanner(System.in);
```

```

String str = in.nextLine();
if (str.length() < 7) System.out.println(0);
String str1 = str.substring(2, str.length() - 2);
String[] splits = str1.split("\\[", "\\]");
List<int[]> list = new ArrayList<>();
for (String split : splits) {
    String[] cur = split.split(",");
    list.add(new int[] {Integer.parseInt(cur[0]), Integer.parseInt(cur[1])});
}
//每个箱子能够装多少,然后找一个能装下他的最小的箱子
list.sort((a, b) -> a[0] - b[0]);
int L = list.get(0)[0];
int W = list.get(0)[1];
int res = 1;
for (int i = 1; i < list.size(); i++) {
    int[] curArr = list.get(i);
    if (curArr[0] > L && curArr[1] > W) {
        res++;
        L = curArr[0];
        W = curArr[1];
    }
}

int res2 = 1;
list.sort((a, b) -> a[1] - b[1]);
L = list.get(0)[0];
W = list.get(0)[1];
for (int i = 1; i < list.size(); i++) {
    int[] curArr = list.get(i);
    if (curArr[0] > L && curArr[1] > W) {
        res2++;
        L = curArr[0];
        W = curArr[1];
    }
}

System.out.println(Math.max(res, res2));
}
}

```