

452. Minimum Number of Arrows to Burst Balloons

Add to List

QuestionEditorial Solution

[My Submissions](#)

- Total Accepted: **3334**
- Total Submissions: **8011**
- Difficulty: **Medium**
- Contributors: [abhijeeg](#)

There are a number of spherical balloons spread in two-dimensional space. For each balloon, provided input is the start and end coordinates of the horizontal diameter. Since it's horizontal, y-coordinates don't matter and hence the x-coordinates of start and end of the diameter suffice. Start is always smaller than end. There will be at most 10^4 balloons. An arrow can be shot up exactly vertically from different points along the x-axis. A balloon with x_{start} and x_{end} bursts by an arrow shot at x if $x_{\text{start}} \leq x \leq x_{\text{end}}$. There is no limit to the number of arrows that can be shot. An arrow once shot keeps travelling up infinitely. The problem is to find the minimum number of arrows that must be shot to burst all balloons.

```
Input:
[[10,16], [2,8], [1,6], [7,12]]

Output:
2
```

Explanation:

One way is to shoot one arrow for example at $x = 6$ (bursting the balloons $[2,8]$ and $[1,6]$) and another arrow at $x = 11$ (bursting the other two balloons).

zzw: 穿串串, 这题贼简单, 先排个序 按照数组的第一个元素排, 得到 $[1, 6]$ $[2, 8]$ $[7, 12]$ $[10, 16]$ 然后记录前一个数组的第二个元素(end) 是不是大于当前数组的第一个元素(start) 如果大于增加arrow数目 $[1, 6]$ $[2, 8]$ 穿一串 $[7, 12]$ $[10, 16]$ 再穿一串

```
public int findMinArrowShots(int[][] points) {
    Arrays.sort(points, (a, b) -> a[1] - b[1]);
    int prev = -1; int count=0;
    for(int[] point:points)
    {
        if(prev==-1 || prev<point[0])
```

```
        {
            count++;
            prev = point[1];
        }
    }
    return count;
}
```