

Charotar University of Science and Technology

Faculty of Technology and Engineering

Devang Patel Institute of Advance Technology and Research

Q-1. There are some spherical balloons taped onto a flat wall that represents the XY-plane. The balloons are represented as a 2D integer array points where $\text{points}[i] = [\text{x}_{\text{start}}, \text{x}_{\text{end}}]$ denotes a balloon whose horizontal diameter stretches between x_{start} and x_{end} . You do not know the exact y-coordinates of the balloons.

Arrows can be shot up **directly vertically** (in the positive y-direction) from different points along the x-axis. A balloon with x_{start} and x_{end} is **burst** by an arrow shot at x if $\text{x}_{\text{start}} \leq x \leq \text{x}_{\text{end}}$. There is **no limit** to the number of arrows that can be shot. A shot arrow keeps traveling up infinitely, bursting any balloons in its path.

Given the array points, return *the **minimum** number of arrows that must be shot to burst all balloons.*

Example 1:

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

Output: 2

Explanation: The balloons can be burst by 2 arrows:

- Shoot an arrow at x = 6, bursting the balloons [2,8] and [1,6].
- Shoot an arrow at x = 11, bursting the balloons [10,16] and [7,12].

Example 2:

Input: points = [[1,2],[3,4],[5,6],[7,8]]

Output: 4

Explanation: One arrow needs to be shot for each balloon for a total of 4 arrows.

Example 3:

Input: points = [[1,2],[2,3],[3,4],[4,5]]

Output: 2

Explanation: The balloons can be burst by 2 arrows:

- Shoot an arrow at x = 2, bursting the balloons [1,2] and [2,3].
- Shoot an arrow at x = 4, bursting the balloons [3,4] and [4,5].

Constraints:

- $1 \leq \text{points.length} \leq 10^5$
- $\text{Points}[i].\text{length} == 2$

- $-231 \leq x_{\text{start}} < x_{\text{end}} \leq 2^{31} - 1$

CODE:

```
def minArrow(points):
```

```
    points.sort(key=lambda x: x[0])
```

```
    arrow = 0
```

```
    last = -1
```

```
    for start, end in points:
```

```
        if start > last:
```

```
            arrow += 1
```

```
            last = end
```

```
    return arrow
```

```
x = [[10,16],[2,8],[1,6],[7,12]]
```

```
y = [[1,2],[3,4],[5,6],[7,8]]
```

```
z = [[1,2],[2,3],[3,4],[4,5]]
```

```
print('Number of minimum arrows needed to burst the set1 balloons are:',minArrow(x))
```

```
print('Number of minimum arrows needed to burst the set2 balloons are:',minArrow(y))
```

```
print('Number of minimum arrows needed to burst the set3 balloons are:',minArrow(z))
```

OUTPUT:

```
PS C:\Users\HP\Desktop\clg\SEM5\competitive programming> & C:\Users\HP\Desktop\clg\SEM5\competitive programming\cie2.py"
Number of minimum arrows needed to burst the set1 balloons are: 2
Number of minimum arrows needed to burst the set2 balloons are: 4
Number of minimum arrows needed to burst the set3 balloons are: 2
PS C:\Users\HP\Desktop\clg\SEM5\competitive programming> █
```