75 Days of Code
Day 73
Problem no : Leetcode 452
Problem Title . Minimum Number of Arrows to Burst Balloons
Problem type : Intervals

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 points[i] = [xstart, xend] denotes a balloon whose horizontal diameter stretches between xstart and xend. 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 xstart and xend is burst by an arrow shot at x if xstart <= x <= xend. 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, bu

.

```typescript
function findMinArrowShots(points: number[][]): number {
  if (!points.length) {
    return 0;
  }
  points.sort((a, b) => a[1] - b[1]);

  let arrows = 1;
  let prevEnd = points[0][1];

  for (let i = 1; i < points.length; i++) {
    const [start, end] = points[i];
    // If the current balloon starts after the previous one ends,
    // you need to shoot another arrow.
    if (start > prevEnd) {
      arrows++;
      prevEnd = end; // Update the ending point of the current arrow
    }
  }
  return arrows;
}
```

Accepted

Runtime	Details	Memory
201 ms		73.34 MB
Beats 50.85% of users with TypeScript		Beats 55.93% of users with TypeScript

More challenges