Find First and Last Position of Element in Sorted Array

For this problem we need an $O(\log n)$ algorithm, so binary search is required.

Approach: Two Binary Searches:

We perform two separate binary searches:

1. Find the first occurrence (leftmost index) of target.
2. Find the last occurrence (rightmost index) of target.

If target is not found, return $[-1, -1]$.

Binary Search for First Occurrence:

- While searching, if nums[mid] == target, move left to find earlier occurrences.

Binary Search for Last Occurrence:

- While searching, if nums[mid] == target, move right to find later occurrences.

Complexity:

- Time: $O(\log n)$ for each binary search $\rightarrow O(\log n)$ total.
- Space: $O(1)$