Search Insert Position

Since the array is sorted and we need $O(\log n)$ runtime, we use binary search.

Algorithm:

1. Initialize left = 0, right = nums.size() - 1.
2. Perform standard binary search:
   - If nums[mid] == target, return mid.
   - If nums[mid] < target, move left = mid + 1.
   - Else move right = mid - 1.
3. If not found, return left (correct insertion point).

Complexity:

- Time: $O(\log n)$
- Space: $O(1)$