Strivers-A2Z-DSA-Sheet-main\02.Binary Search\1D Arrays\06.First_and_last_position.cpp

```cpp
/*
QUESTION:
Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value.

If target is not found in the array, return [-1, -1].

You must write an algorithm with O(log n) runtime complexity.

Example:

Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]

APPROACH:
1. Use lower_bound to find the index of the first occurrence of the target in the array.
2. If the target is not found, return [-1, -1].
3. Use upper_bound to find the index of the last occurrence of the target in the array.
4. Return the range [first, last-1] as the starting and ending positions.

CODE:
*/

vector<int> searchRange(vector<int>& nums, int target) {
    int first = lower_bound(nums.begin(), nums.end(), target) - nums.begin();
    // if the target is not found, return [-1, -1]
    if (first == nums.size() || nums[first] != target)
        return {-1, -1};
    int last = upper_bound(nums.begin(), nums.end(), target) - nums.begin();
    return {first, last-1};
}

// TIME COMPLEXITY: O(log n)
// SPACE COMPLEXITY: O(1)

```