```
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

public class Program {
  // O(log(n)) time | O(log(n)) space
  public static int[] SearchForRange(int[] array, int target) {
    int[] finalRange = {-1, -1};
    alteredBinarySearch(array, target, 0, array.Length - 1, finalRange, true);
    alteredBinarySearch(array, target, 0, array.Length - 1, finalRange, false);
    return finalRange;
  }

  public static void alteredBinarySearch(int[] array, int target, int left, int right,
    int[] finalRange, bool goLeft) {
    if (left > right) {
      return;
    }
    int mid = (left + right) / 2;
    if (array[mid] < target) {
      alteredBinarySearch(array, target, mid + 1, right, finalRange, goLeft);
    } else if (array[mid] > target) {
      alteredBinarySearch(array, target, left, mid - 1, finalRange, goLeft);
    } else {
      if (goLeft) {
        if (mid == 0 || array[mid - 1] != target) {
          finalRange[0] = mid;
        } else {
          alteredBinarySearch(array, target, left, mid - 1,
            finalRange, goLeft);
        }
      } else {
        if (mid == array.Length - 1 || array[mid + 1] != target) {
          finalRange[1] = mid;
        } else {
          alteredBinarySearch(array, target, mid + 1, right,
            finalRange, goLeft);
        }
      }
    }
  }
}

```