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

#include <vector>
using namespace std;

vector<int> searchForRange(vector<int> array, int target);
void alteredBinarySearch(vector<int> array, int target, int left, int right,
                         vector<int> *finalRange, bool goLeft);

// O(log(n)) time | O(log(n)) space
vector<int> searchForRange(vector<int> array, int target) {
  vector<int> finalRange{-1, -1};
  alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange, true);
  alteredBinarySearch(array, target, 0, array.size() - 1, &finalRange, false);
  return finalRange;
}

void alteredBinarySearch(vector<int> array, int target, int left, int right,
                         vector<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->at(0) = mid;
      } else {
        alteredBinarySearch(array, target, left, mid - 1, finalRange, goLeft);
      }
    } else {
      if (mid == array.size() - 1 || array[mid + 1] != target) {
        finalRange->at(1) = mid;
      } else {
        alteredBinarySearch(array, target, mid + 1, right, finalRange, goLeft);
      }
    }
  }
}

```