binarySearch

```cpp
//
// Created by yjs on 2022/4/21.
//

#include <bits/stdc++.h>

using namespace std;


class Search {
public:

    int binarySearch(const vector<int> nums, const int &target) {
        int left = 0, right = nums.size() - 1;
        while (left <= right) {

            int mid = left + (right - left) / 2;

            if (nums[mid] == target) {
                return mid;
            } else if (nums[mid] < target) {
                left = mid + 1;
            } else if (nums[mid] > target) {

                right = mid - 1;

            }


        }
        return -1;

    }


    int lowerSearch(const vector<int> nums, const int &target) {
        // [a,b)
        int left = 0, right = nums.size();
        while (left < right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] == target) {
                right = mid;
            } else if (nums[mid] < target) {
                left = mid + 1;
            } else if (nums[mid] > target) {
                right = mid;
            }
        }
        return left;
    }
```

```
    int upperSearch(const vector<int> nums, const int &target) {
        // (a,b]
        int left = 0, right = nums.size();
        while (left < right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] == target) {
                left = mid + 1;
            } else if (nums[mid] < target) {
                left = mid + 1;
            } else if (nums[mid] > target) {
                right = mid;
            }
        }
        return left - 1;
    }


};


int main() {

    vector<int> nums{1, 25, 69, 25, 17, 17, 14};
    sort(nums.begin(), nums.end());
    for_each(nums.begin(), nums.end(), [](auto c) { cout << c << " "; });
    cout << endl;
    Search search1;

    cout << search1.binarySearch(nums, 17) << endl;
    cout << search1.lowerSearch(nums, 17) << endl;
    cout << search1.upperSearch(nums, 17) << endl;


    return 0;
}
```