

# CSE 2020 Computer Science II

Module 6.2 – Searching, Empirical Study

Instructor: Kerstin Voigt

School of CSE, CSUSB

```
// return index at which target found, if not found return -1;

template <typename T>
int linear_search_V(const Vector<T>& vec, const T& target,
                   int& ops)
{
    ops = 0;
    for (int i = 0; i < vec.size(); i++)
    {
        ops++;
        if (vec[i] == target)
            return i;
    }
    return -1; // not found
}
```

```
// return "position number" (0 lowest) at which target
// found in lst; else return position number -1;

template <typename T>
typename List<T>::const_iterator linear_search_L(
    const List<T>& lst, const T& target, int& ops)
{
    ops = 0;

    typename List<T>::const_iterator itr;
    for (itr = lst.begin(); itr != lst.end(); ++itr)
    {
        ops++;
        if (*itr == target)
            return itr;
    }
    return lst.end();
}
```

```

// binary search of Vector;

template <typename T>
int binary_search_V(const Vector<T>& vec, const T& target,
                    int& ops)
{
    ops = 0;
    int low = 0;
    int high = vec.size() - 1;

    while (low <= high)
    {
        ops++;
        int mid = (low + high) / 2;

        if (vec[mid] < target)
            low = mid + 1;    // next: search upper half
        else if (vec[mid] > target)
            high = mid - 1;   // next: search lower half
        else
            return mid;      // middle item is target; return mid
    }
    return -1;
}

```

```
#include <iostream>
#include <algorithm>
#include "Vector.h"
#include "List.h"
#include "Random.h"
#include "Searching.h"
using namespace std;

int main()
{
    for (int k = 20; k <= 100; k += 20)
    {
        Vector<int> myvec;
        random_vector_norep(k, 1, 1000, myvec, 0);

        Vector<int> sortvec(myvec);
        sort(sortvec.begin(), sortvec.end());

        List<int> sortlist
        for (int i = 0; i < sortvec.size(); i++)
            sortlist.push_back(sortvec[i]);

        Vector<int> five_targets;
        random_vector_norep(5, 0, myvec.size()-1,
                           five_targets, 0);

        int ops1 = 0;
        int ops2 = 0;
        int ops3 = 0;
```

```

for (int i = 0; i < 5; i++)
{
    int target = myvec[five_targets[i]];
    linear_search_V(sortvec, target, ops1);
    linear_search_L(sortlist, target, ops2);
    binary_search_V(sortvec, target, ops3);

    cout << endl;
    cout << "Searching linear structure of size "
        << myvec.size()
        << " for target " << target << ":"
        << endl << endl;
    cout << "linear_search_V: " << ops1
        << " ops" << endl;
    cout << "linear_search_L: " << ops2
        << " ops" << endl;
    cout << "binary_search_V: " << ops3
        << " ops" << endl;

    cout << endl;
}
}
return 0;
}

```

Searching linear structure of size 20 for target 216:

linear\_search\_V: 4 ops

linear\_search\_L: 4 ops

binary\_search\_V: 5 ops

Searching linear structure of size 20 for target 328:

linear\_search\_V: 6 ops

linear\_search\_L: 6 ops

binary\_search\_V: 4 ops

Searching linear structure of size 20 for target 206:

linear\_search\_V: 3 ops

linear\_search\_L: 3 ops

binary\_search\_V: 4 ops

Searching linear structure of size 20 for target 76:

linear\_search\_V: 1 ops

linear\_search\_L: 1 ops

binary\_search\_V: 4 ops

Searching linear structure of size 20 for target 510:

binary\_search\_V: 7 ops

Searching linear structure of size 100 for target 146:

linear\_search\_V: 17 ops

linear\_search\_L: 17 ops

binary\_search\_V: 7 ops

Searching linear structure of size 100 for target 630:

linear\_search\_V: 65 ops

linear\_search\_L: 65 ops

binary\_search\_V: 5 ops

Searching linear structure of size 100 for target 379:

linear\_search\_V: 40 ops

linear\_search\_L: 40 ops

binary\_search\_V: 5 ops

Searching linear structure of size 100 for target 813:

linear\_search\_V: 83 ops

linear\_search\_L: 83 ops

binary\_search\_V: 7 ops



```
lab9_output2 - Notepad
File Edit Format View Help

Linear Search of Vector:

      1  2  3  4  5
-----
50  18  7  21  29  38
100 34 100 38  94  76
150 66 149 63  51 116
200 36  60 190 191 150
250 82 174 239 240 207

Linear Search of List:

      1  2  3  4  5
-----
50  18  7  21  29  38
100 34 100 38  94  76
150 66 149 63  51 116
200 36  60 190 191 150
250 82 174 239 240 207

Binary Search of Vector:

      1  2  3  4  5
-----
50  3  5  4  5  2
100 5  7  6  4  6
150 7  7  7  5  7
200 8  7  8  6  2
250 8  7  6  8  6

Ln 209, Col 1  100%  Windows (CRLF)  UTF-8
```

*Possible with some additional coding ...*

## Upcoming Homework Assignment

- ☐ Produce your own empirical study of linear vs. binary search
- ☐ Report, plot and discuss findings

\*\*\* End of Module 6.2 \*\*\*