

CEN235 – DATA STRUCTURES

Fall – 2018

Lab 6: Linear (Sequential) Search and Binary Search Algorithms

1. Linear (Sequential) Search: Write a C program to run linear search. Create a 2-D Integer array: `int myarray[M][N]`, where M is always 100 (should be defined at the top using `#define` keyword) and N has four different values; 1.000, 10.000, 100.000 and 1.000.000 (should be defined at the top using `#define` keyword). You must fill these arrays with random values between $-N$ and $+N$. Create a random number between $-N$ and $+N$ to be searched in the given lists for each test. As you have always 100 lists for each test, you should fill the following table for each test separately (You can see my results in red). [50pts]

	N=1.000	N=10.000	N=100.000	N=1.000.000
Find percentage in tests	0.23	0.86	1.0	1.0
Min index found	45	38	26402	18900
Max index found	958	9969	36302	28800
Total Program Execution Time in sec.	0.016s	0.991s	5.337s	1.597s

Could you allocate memory? Did you have any problem like stack overflow? How do you handle this problem? What about the results and percentages and execution times?

2. Binary Search: In this part of the Lab assignment, you are expected to write a C program to read the dictionary (a text file having one word in each line) and search for a word given by the user in the dictionary. There are **370099 words** in the dictionary provided. If you find the word, you have to give the number of steps in finding the desired word and the line number of the word using binary search algorithm. You can use the following binary search algorithm for the strings: [50pts]

```
void binary_search(char *list_of_words[], int size, char *target){
    int bottom= 0;
    int mid;
    int top = size - 1;

    while(bottom <= top){
        mid = (bottom + top)/2;
        if (strcmp(list_of_words[mid], target) == 0){
            printf("%s found at location %d.\n", target, mid+1);
            return;
        } else if (strcmp(list_of_words[mid], target) > 0){
            top = mid - 1;
        } else if (strcmp(list_of_words[mid], target) < 0){
            bottom = mid + 1;
        }
    }
}
```