Julius Ranoa

CSC 121 001 Computer Science I

Homework – Chapter 9 Searching, Sorting, and Algorithm Analysis

**Part I. Review Questions @ Page 639 – 640.**

Qn. 1 – 6, 11 – 18.

1. The *linear* search algorithm steps sequentially through an array, comparing each item with the search value.

2. The *binary* search algorithm repeatedly divides the portion of an array being searched in half.

3. The *linear* search algorithm is adequate for small arrays but not for large arrays.

4. The *binary* search algorithm requires that the array’s contents be sorted.

5. The *average* number of comparisons performed by linear search to find an item in an array of N elements is *N / 2*.

6. The *maximum* number of comparisons performed by linear search to find an item in an array of N elements is *N*.

11. If an array is sorted in *ascending* order, the values are stored from lowest to highest.

12. If an array is sorted in *descending* order, the values are stored from highest to lowest.

13. Bubble sort places *one* number(s) in place on each pass through the data.

14. Selection sort places *one* number(s) in place on each pass through the data.

15. To sort N numbers, bubble sort continues making passes through the array until *no passes are made*.

16. To sort N numbers, selection sort makes *N – 1* passes through the data.

17. Why is selection sort more efficient than bubble sort on large arrays?  *Selection sort does fewer data swaps than the bubble sort.*

18. Which sort, bubble sort or selection sort, would require fewer passes to sort a set of data that is already in the desired order? *Bubble Sort.*

**Part II. Programming Challenge @ Page 644.**

Qn. 15 – Using Files: String Selection Sort Modification.

Screenshot of Runtime: 

Source Code:

1. *NameList.h*
2. *NameList.cpp*
3. *main.cpp*

The source code is also stored at Github.

Link below:

<https://github.com/TheLoneWoof1102/FA17_CSC121001/tree/master/Source%20Code/Homework-Ch9.Qn15>

**main.cpp**

****

***// END of main.cpp.***

**NameList.h**

****

***// END of NameList.h.***

**NameList.cpp**

****

**// *END of NameList.cpp***