**HOMEWORK 1 – Searching, Sorting, and Algorithm Efficiency**

Answer the following questions. **THIS ASSIGNMENT WILL BE DUE AT A DATE TO BE ANNOUNCED IN CLASS.**

1. Briefly describe how the binary search algorithm works. Compare it to the operation of the linear search algorithm. Give one advantage and one disadvantage of the binary search algorithm relative to the linear search algorithm. (10 points)

The binary search algorithm works by repeatedly dividing a a sorted array to find a specific value. The algorithm first takes the middle element and if the element matches the specific value, the algorithm stops if it doesn’t match it checks to see, if the specific element is greater or less than the middle element. If less than it continues the same process in the lower half and also vice versa. Comparing it to the linear search algorithm which searches the entire array sequentially or one by one in order, an advantage of the binary search is immediately found. The binary search algorithm is significantly faster than the linear search algorithm especially the more elements there are in an array. A disadvantage is that the binary search algorithm only works when the array you are given is sorted while the linear search can be used on any array.2. Given the list declaration vals = [16, 3, 8, 9, 2, 15, 6]. Draw or otherwise indicate the contents of the list after the second comparison using the selection sort to sort the list in descending order. Do the same for the insertion sort after three items have been placed in their final positions. (20 points)

After the second comparison using a selection sort the contents of the list would be = [16, 15, 8, 9, 2, 3, 6]. The way the selection sort works is by finding the, in this case, the largest elements and placing it in index 0. Then at index 1 it goes through the list again finding the largest element excluding the element it previously found. This process is repeated until the list is sorted. For the insertion sort after three items have been placed the list would be as follows = [16, 15, 8, 9, 2, 3, 6]. The three elements in the final positions would be 16, 15, and 3. The insertion sort starts at the first elements and placed the next element either before or after the first element creating a ‘sublist’. The next element would be placed in its proper place among this ‘sublist’. This process is repeated until the everything is sorted.

3. Given the list from Problem 2. Draw or otherwise indicate the contents of the list after the 3rd comparison of the fourth pass through the list using the bubble sort to sort the list in ascending order. (10 points)

Bubble sort is a sorting algorithm which compares elements next to each other and swaps them if they are out of order. This repeats for the next pair and, it does this repeatedly until the list is sorted. For the list from problem 2 = [16, 3, 8, 9, 2, 15, 6], after the different pass the list would look like

Pass 1 = [3, 8, 9, 2, 15, 6, 16]

Pass 2 = [3, 8, 2, 9, 6, 15, 16]

Pass 3 = [3, 2 , 8, 6, 9, 15, 16]

Pass 4 (3rd Comparison) = [2, 3, 6, 8, 9, 15, 16]

4. Given the code below.

# Dr. Z - February 12-13, 2002 - Program I from old Test I

# modified from C++ to Python on Sunday, February 6, 2022.

i = j = k = n = x = y = z = 1

n = int(input("Please enter any positive integer value: "))

if(n<1):

n = -n

for i in range(n):

print("3\*4 = ",3\*4,".",sep='')

for j in range(n):

x = y\*x

y = x\*y + z

for k in range(1,n-1):

print(i,j,k)

z = z\*y\*z

* 1. Formulate the function f(n) of the number of multiplications in this program as a function of n. (10 points)

The function f(n) would be = 2n^3 - 2n^2 + n

* 1. **EXTRA CREDIT!!!** Formulate the function c(n) of the number of comparisons in this program as a function of n. (10 points)

The function c(n) of the number of comparisons would be -> c(n) = 1