# **CS 160 LAB 7**

# Chapter 2, Chapter 3

# Algorithm Discovery and Design and The Efficiency of Algorithms

1. Write an algorithm that uses a loop
   1. to input 10 pairs of numbers, where each pair represents the score of a football game with the Computer State University (CSU) score listed first, and
   2. for each pair of numbers, to determine whether CSU won or lost.

After reading in these 10 pairs of values, print the won/lost/tie record of CSU. In addition, if this record is a perfect 10-0, then print the message ‘Congratulations on your undefeated season’.

BEGIN  
int i=0  
int CSU[10]  
int op[10]  
int count=0  
for i<10 do  
READ CSU[i],op[10]  
if CSU[i]>op[i] then  
DISPLAY "CSU won"  
count=count+1  
end if  
if CSU[i]==op[i] then  
DISPLAY "its a tie"  
end if  
if CSU[i]<op[i] then  
DISPLAY "Its a DEFEAT"  
end if  
i++  
end for  
if count==10 then  
DIPLAY Congratulations on your undefeated season  
end if  
END

1. Use the Find Largest algorithm of [Figure 2.14](javascript://) to help you develop an algorithm to find the median value in a list containing N unique numbers. The median of N numbers is defined as the value in the list in which approximately half the values are larger than it and half the values are smaller than it. For example, consider the following list of seven numbers.

* 26, 50, 83, 44, 91, 20, 55

The median value is 50 because three values (20, 26, and 44) are smaller and three values (55, 83, and 91) are larger. If N is an even value, then the number of values larger than the median will be one greater than the number of values smaller than the median.

-Step 1: Get the values A1, A2,…….,An of the numbers in the list.

-Step 2: If “n” is even, then

Let Median = n / 2

Else

Let Median = (n+1) / 2

-Step 3: While (n >= Median), do steps 4 through 9.

-Step 4: Use find Largest Algorithm to find the largest number location in the list A1, A2,…..An

-Step 5: Exchange A ValueLocation and An as follows

-Step 6: Assign the value of “An” to variable “S”

-Step 7: Assign the value of “A ValueLocation” to varianle “An”.

-Step 8: Assign the value of “S” to variable “A ValueLocation”.

-Step 9: Decrement the value of “n” by 1 and then assign to “n”.

-Step 10: Display the value of median “A Median”

-Step 11: Stop.

1. On the sixth line of the Find Largest algorithm of Figure 2.14, there is an instruction that reads,   
    While (i ≤ n) do  
   Explain exactly what would happen if we changed that instruction to read as follows:
2. While (i ≥ n) do
3. While (i < n) do
4. While (i = n) do
5. The sixth line of the “find largest algorithm” is changed by the above while() statement.

-If the value of the size list (n) is less than or equal to 2, it generates the infinite loop.

-If the value of the size list (n) is greater than 2, it exits the loop.

b) The sixth line of the “find largest algorithm” is changed by the above while() loop.

-If condition satisfies, it checks for the n-1 value in the list.

-The above condition doesn’t check the last element in the list.

c) The sixth line of “find largest algorithm” is changed by the above while() statement.

-The above condition satisfies only if the size of the list is 2. So, it compares the first value

with the second value.

1. Suppose you have a series of 10 storage locations named S[1] , S[2], etc. thru S[10]. Using a while loop, an index variable K, and 1 temporary storage location named T, write an algorithm that will reverse the order of values in the storage locations so that S[1] holds the value formerly in S[10], S[2] holds the value formerly in S[9], etc., and S[10] holds the value formerly in S[1]. Use the index variable or an expression inside the square brackets of S[ ] to indicate which storage location is being accessed.
2. Rewrite the algorithm for problem 8 (the reverse-the-storage-locations problem) so that it will work properly with N storage locations, where N is any positive integer.
3. Here is a list of seven names:

Sherman, Jane, Ted, Elise, Raul, Maki, John

Search this list for each name in turn, using sequential search and counting the number of comparisons for each name. Now take the seven comparison counts and find their average. Did you get a number that you expected? Why?

- Starting from the left, traverse the list, comparing each element with "target":

- If they are equal, return the index of the current element.

- If they are not equal, move to the next element.

- Return -1

Let N = {'Sherman', 'Jane', 'Ted', 'Elise', 'Raul', 'Maki', 'John'} be the list we are working on

1. Searching for 'Sherman': 'Sherman' is found on the first comparision itself. So number of comparisons = 1.
2. Searching for 'Jane': 'Jane' is found after comparing "target"( = 'Jane') to 'Sherman' and 'Jane'. So 2comparisions.
3. Searching for 'Ted': 'Ted' is found after comparing "target"( = 'Ted') to 'Sherman', Jane' and 'Ted'. So, 3comparisons.
4. Searching for 'Elise: 'Elise is found after 4 comparisons.
5. Searching for 'Raul': 'Raul' is found after 5 comparisons.
6. Searching for 'Maki': 'Maki' is found after 6 comparisons.
7. Searching for 'John': 'John' is found after 7 comparisons.

So, average number of comparisions that were made = (1+2+3+4+5+6+7) / 7 = 4.

1. Perform a selection sort on the list 7, 4, 2, 9, 6. Show the list after each exchange that has an effect on the list ordering.

-Step 1:

Swap 7 and 2

2 4 7 9 6

-Step 2:

Swap 7 and 6

2 4 6 9 7

-Step 3:

Swap 9 and 7

2 4 6 7 9

Therefore, the sorted list element after the selection sort is 2, 4, 6, 7, 9.