Course Title: CS350 Student Name: Sean Poston SEMO ID#: S01882786
Student Email: sposton1s@semo.edu Date: 9/25/2021

Question 1:

```
ARRAYMAX(A, N)
    a. CURRENTMAX <- A[0]
    b. FOR I<-1 TO N-1 DO
    c. IF A[I] > CURRENTMAX THEN
        i. CURRENTMAX <- A[I]
    d. {INCREMENT COUNTER I}
    e. RETURN CURRENTMAX
```

Looking at this code segment, we can see that the only loop is on line b. We see that the loop goes for the length of the array, and everything else is constant, so the time complexity would be **O(n)**.

Question 2:

```
PREFIXAVERAGES1(X,N)
    a. INPUT: ARRAY X OF N INTEGERS
    b. OUTPUT: ARRAY A OF PREFIX AVERAGES OF X
    c. A <- NEW ARRAY OF N INTEGERS
    d. FOR I <- 0 TO N – 1 DO
        i. S <- X[0]
        ii. FOR J <- 1 TO I DO
            1. S <- S + X[J]
            2. A[I] <- S/(I+1)
    RETURN A
```

This code segment has two loops. The outer loop on line c loops the length of the array. The inner loop only goes to whatever the current place of i is. This means that in the *worst case*, which is what big O represents, it will also loop the full complement of the array. This means that this algorithm is **O(n^2)**.

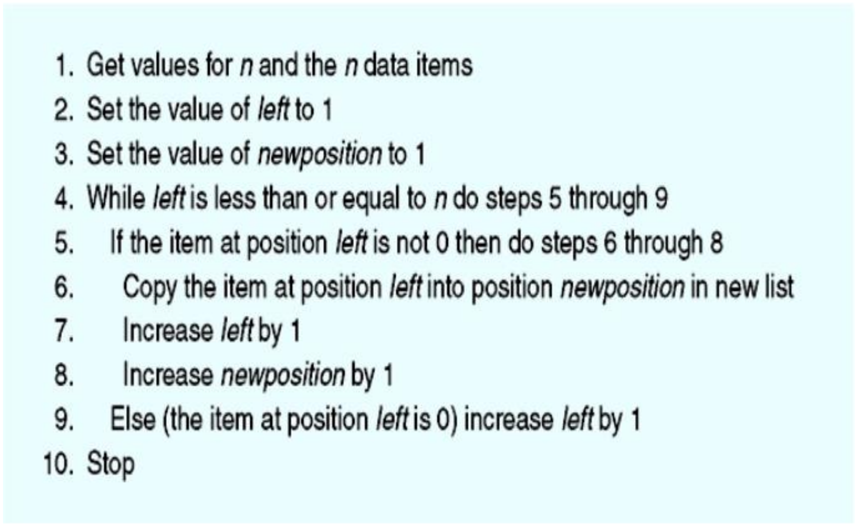
Question 3:

```
PREFIX AVERAGES2(X,N)
    INPUT: ARRAY X OF N INTEGERS
    OUTPUT: ARRAY A OF PREFIX AVERAGES OF X
    A <- NEW ARRAY OF N INTEGERS
    S<- 0
    FOR I<- 0 TO N-1 DO
        S<-S + X[I]
        A[I] <- S/(I+1)
    RETURN A
```

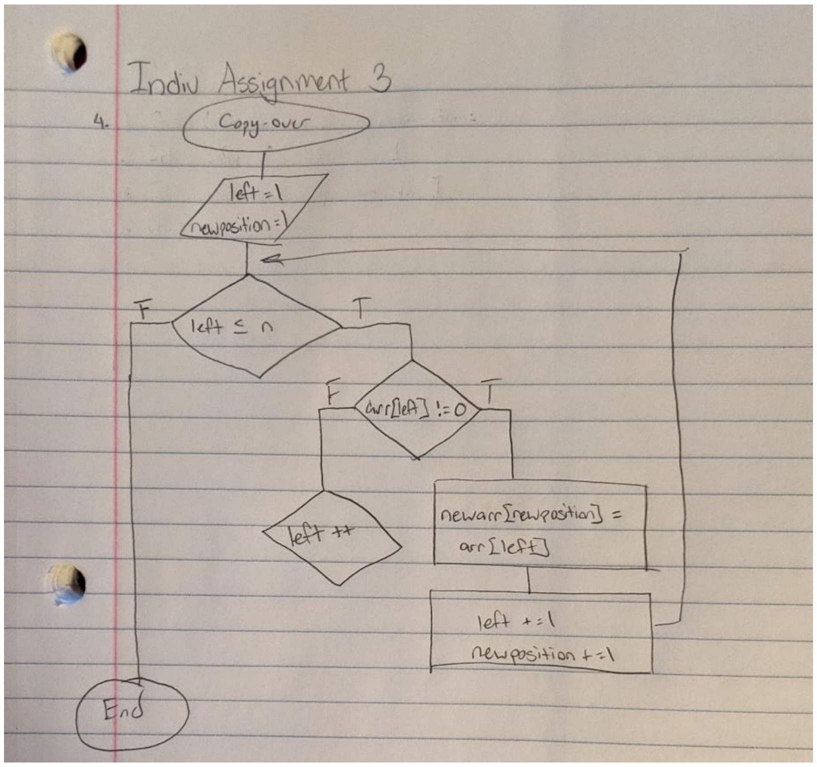
This segment only has one loop that spans the length of the array, so it’s **O(n)**.

Question 4:

Please draw a flowchart to explain the Copy-over algorithm for data cleanup.

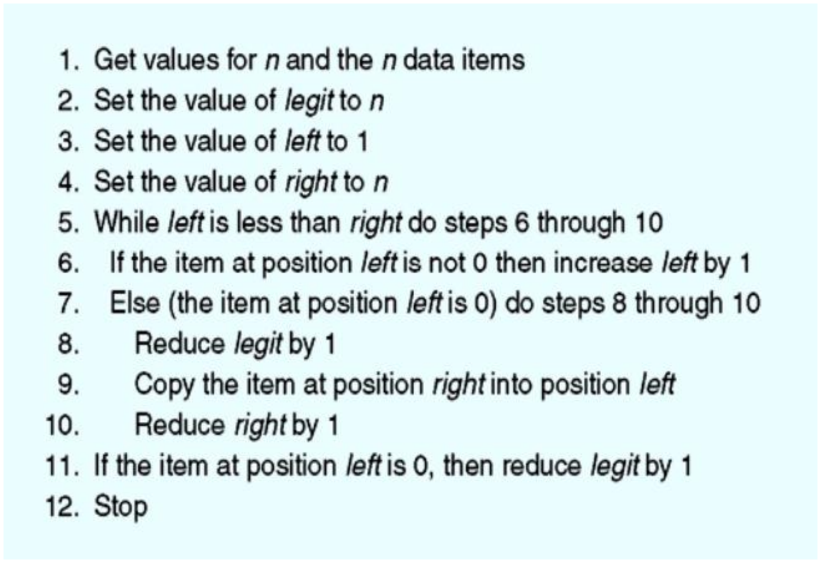


```
1. Get values for n and the n data items
2. Set the value of left to 1
3. Set the value of newposition to 1
4. While left is less than or equal to n do steps 5 through 9
5.    If the item at position left is not 0 then do steps 6 through 8
6.      Copy the item at position left into position newposition in new list
7.      Increase left by 1
8.      Increase newposition by 1
9.    Else (the item at position left is 0) increase left by 1
10. Stop
```

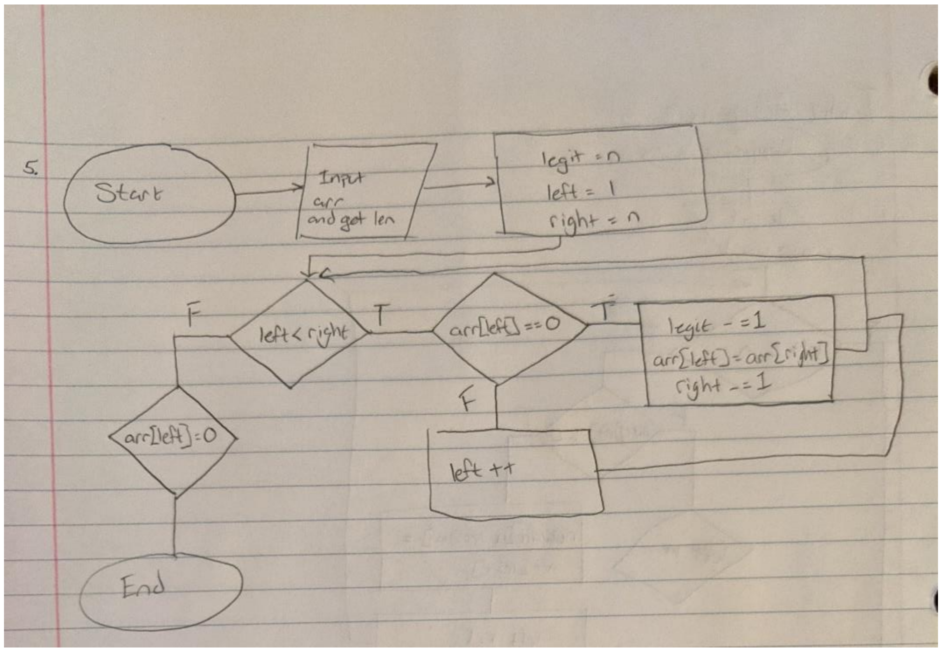


Question 5:

Please draw a flowchart to explain the Converging-pointers algorithm for data cleanup.



```
1. Get values for n and the n data items
2. Set the value of legit to n
3. Set the value of left to 1
4. Set the value of right to n
5. While left is less than right do steps 6 through 10
6.   If the item at position left is not 0 then increase left by 1
7.   Else (the item at position left is 0) do steps 8 through 10
8.      Reduce legit by 1
9.      Copy the item at position right into position left
10.     Reduce right by 1
11. If the item at position left is 0, then reduce legit by 1
12. Stop
```


5.
Start
Input arr and get len
legit = n
left = 1
right = n
F
left < right
T
arr[left] == 0
T
legit - = 1
arr[left] = arr[right]
right -= 1
F
left ++
arr[left] = 0
End