**CSC 180, Lab #8**

**Fall 2024**

***Directions:*** Turn in a hard copy of this assignment, with your answers written on this or another sheet of paper, as well as the completed Jupyter Notebook that goes with this lab.

1. In class, we learned that a formula for adding the integers between 1 and *n* is given by the following: 𝑛(𝑛 + 1)/2. Using this formula [5 points]
   1. Find the sum of the integers between 1 and 50
   2. Find the sum of the integers between 1 and 70
2. The code below contains two algorithms for determining whether a word is in a list of words [9 points]

set wordFound to False for w in words :

if w = word :

set wordFound to True

if wordFound = True :

print 'word found!'

Assume that the length of the word list is *n*.

1. Assuming that the word is in the list, how many operations are required by the algorithm? Note: Count the following as single operations: assignment, comparison, and printing; the statement *for w in words* should count as a three operations, which is repeated *n* times.
2. What is the order of magnitude of this algorithm, using Big Theta notation?
3. What is the space requirement for this algorithm (not including the list of words)?
4. Suppose an algorithm with an input of size *n = 10* takes 3 *ms* to run.
   1. If the running time is θ(n), approximately how long would it take if *n* = 20?
   2. If the running time is θ(1), approximately how long would it take if *n* = 20?
   3. If the running time is θ(n2), approximately how long would it take if *n* = 20?
5. Consider the list containing 3, 0, 1, 0, and 7, and complete the table below to specify the values of *position*, *num\_valid* and the list after each iteration of the *while* loop in the *shuffle left* algorithm. [6 points]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | position | num\_valid | List | | | | |
| Original list | 0 | 5 | 3 | 0 | 1 | 0 | 7 |
| After iteration 1 | 1 |  |  |  |  |  |  |
| After iteration 2 | 1 |  |  |  |  |  |  |
| After iteration 3 | 2 |  |  |  |  |  |  |
| After iteration 4 | 2 |  |  |  |  |  |  |
| After iteration 5 (final) | 3 |  |  |  |  |  |  |

1. Consider the list containing 3, 0, 1, 0, and 7, and complete the table below to specify the value of the *copyNum* listafter each iteration of the *for* loop in the *copyOver* algorithm. [6 points]

|  |  |  |  |
| --- | --- | --- | --- |
| copyNum | - | - | - |
| After iteration 1 |  |  |  |
| After iteration 2 |  |  |  |
| After iteration 3 |  |  |  |
| After iteration 4 |  |  |  |
| After iteraton 5 |  |  |  |

1. Consider the list containing 3, 0, 1, 0, and 7, and complete the table below to specify the values of *left*, *right*, *num\_valid*, and the *list* after each iteration of the *while* loop in the *converging pointers* algorithm. [6 points]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Left | Right | num\_valid | List | | | | |
| Original list | 0 | 4 | 5 | 3 | 0 | 1 | 0 | 7 |
| After iteration 1 |  |  |  |  |  |  |  |  |
| After iteration 2 |  |  |  |  |  |  |  |  |
| After iteration 3 |  |  |  |  |  |  |  |  |
| After iteration 4 (final) |  |  |  |  |  |  |  |  |

After the final iteration, number[left] is not valid, so *num\_valid* is decreased by 1. The algorithm is now complete. There are 3 valid elements and the first 3 elements of the list should be 3, 7, and 1.