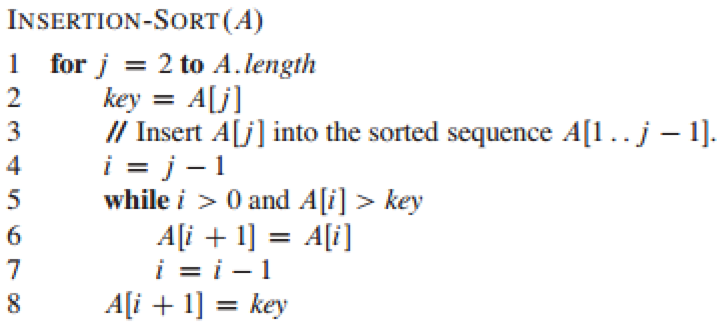
Spring 2023 – CS 303 Algorithms and Data Structures
Lab 2

Objectives:

- Implement insertion sort algorithm
- Evaluate performance of insertion sort with increasing array size

Problems

1. Implement a method that will sort a given array using the insertion sort algorithm (given below).
2. Write a driver program to test the insertion algorithms implemented in Question 1. Read the input file “1000.txt” for the input numbers and store them in an array. Sort this array using insertion sort.
3. Test the program for the different size input files provided in Canvas.
4. Record the runtime for insertion sort on various sized arrays by using the provided files. Comment on how the execution time of insertion sort varies with size of the input array. Use a table or plot to summarize the results and document your observations and explanations in the report.



```
INSERTION-SORT($A$)
for $j = 2$ to $A.length$
    $key = A[j]$
    // Insert $A[j]$ into the sorted sequence $A[1..j-1]$.
    $i = j - 1$
    while $i > 0$ and $A[i] > key$
        $A[i+1] = A[i]$
        $i = i - 1$
    $A[i+1] = key$
```

Note: The above pseudo code assumes that the array indexing is starting from 1. If you are using a programming language that uses array indexing starting from 0, you have to modify the pseudo code accordingly.

Submission:

- You are required to submit a written report (.pdf) and your project file (.zip) to the Canvas
- Homework report must follow the guidelines provided in the sample report uploaded in Canvas. Please include the screenshot of your code and outputs of your code at the end of your report.
- Do not forget to submit Independent Completion Form
- When you create the code to read the file use relative path instead of absolute path

DATA

1000, 2500,5000,10000,25000,50000,100000,250000,500000,1000000

Grading Rubric

Coding	Implementing Algorithms	20 points
	Producing Correct Outputs	20 points
Report	Explaining the algorithms used	10 points
	Displaying the output with a graph or table	20 points
	Comparing the outputs and discussing the time complexity of algorithms	20 points
	Correct submissions of the files (ICF, Code.zip, report.pdf)	10 points