**Programming Assignment 1**

**Due: 3/6/2023**

The objective of this assignment is to explore sorting algorithms and benchmark for the running times.

**Assignment**

1. Implement three programs for three sorting algorithms: **InsertionSort** , **QuickSort** and, **MergeSort**.
2. Each algorithm accepts an array of integer as an input and returns an array of arrays with two elements.

* The first element of the output is a descending sorted array of input
* The second element of the output is the running time of the algorithm in milliseconds.

For example:

input = [2,3,4,1]

output = [[4,3,2,1], [0.03]]

output[0] is descending sorted array and output[1] is the running time of the algorithm in millisecond.

1. Programs use four different inputs, so totally you need to submit 12 different results.
   1. A descending sorted array with 2000 elements
   2. An ascending sorted array with 2000 elements
   3. A randomly generated array with 1000
   4. A randomly generated array with 2000

Notes:

1. Functions that generate arrays above are part of the given templates, so no need to write it again.
2. You don’t need to submit the sorted array as the result.
3. Run your programs for each of scenarios above and record the running time results by adding them to the given excel sheet.
4. This is how your program should be called from the command-line:

c:\> insertionsort.py inputtype, elements\_count ,seed

* **inputtype:** is either a,b,c . Letters refer to the type of your input array described in the previous section.
* **elements\_count:** The number of elements in input array
* **seed:** is used only for the random base inputs that guarantees I get the same result of

random array so I can grade your code based on it. You can change it while testing. I will test your code with the seed value of 2.

Example1:

python insertionsort\_lastname\_firstname.py c, 1000, 2

Use insertion sort to sort a randomly generated array of 1000 elements

Note that, c and d use the same function with different elements count

Example2:

python quicksort\_lastname\_firstname.py a, 2500, 2

Use quick sort to sort a descending sorted array of 2500 elements

1. Run your algorithms for each of four inputs listed in the section 3 above, measure the running time of algorithms.

**What to submit?**

1. Write 3 programs for each algorithm. Name your programs as:
   1. insertionsort\_lastname\_firstname.py
   2. quicksort\_lastname\_firstname.py
   3. mergesort\_lastname\_firstname.py
2. Fill the excel sheet with your time results.
3. Rename the excel sheet as lastname\_firstname.xlsx
4. Rename the code files to insertionsort\_lastname\_firstname.py for example (do the same for other code files respectively)
5. You should follow general software development rules such as proper and sufficient commenting if it is necessary and proper functions and variable names.
6. Do not copy any code from online resources!