

# INFR 2820U: Algorithms and Data Structures

## Assignment-2: Sorting Algorithm Visualization and Simulation

### Background:

Online shopping platforms handle vast amounts of product data, necessitating not just efficient data management but also effective sorting mechanisms. This assignment offers a deep dive into sorting algorithms, requiring students not only to implement these algorithms but also to visualize each step of their algorithm's execution, including an innovative sensory integration task.

### Objective:

To gain a hands-on, in-depth understanding of sorting algorithms, trace their execution, visualize the sorting process, and enhance the understanding through auditory feedback.

### Topics Covered:

- Sorting Algorithms: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort
- Visualization of Sorting Processes
- Detailed Step-by-Step Algorithm Execution

### Learning Outcomes:

- Implement and trace sorting algorithms through detailed visualization.
- Develop a deeper understanding of the iterative process of sorting algorithms.
- Evaluate the time complexities of these sorting algorithms in step-by-step scenarios.

### Tasks:

#### 1. Detailed Visualization of Sorting Processes:

- For the given array, manually show the result of each step or iteration for the following algorithms, without implementing them in code.
- The array to be sorted in increasing order:

Product Id	11	1	30	2	51	6	29	7	67	15	118	4	89	23
------------	----	---	----	---	----	---	----	---	----	----	-----	---	----	----

- Clearly show all iteration when applying the following sorting algorithms on **Product ID**.
  - ✓ Merge Sort (all iterations)
  - ✓ Quick Sort (first iteration, using the first element as a pivot)

#### 2. Sorting Algorithm Simulation Program:

- Write a program in Python/C++ (or the language of your choice) that simulates the **Merge Sort Algorithm**. Please incorporate sound effects for swaps.
- The program must accept an array of  $n$  numbers, then execute the sort algorithm and print out each step.
- The final program should be submitted in a compatible format for online compilation and execution.

**Deliverables: (5 Marks)**

1. Introduction:
  - Provide context and objectives for the detailed sorting algorithm visualization.
2. Detailed Visualization (2+1 Marks):
  - Documentation that includes detailed illustrations showing the output for each iteration of the sorting process for the two specified algorithms.
  - Write a comparison table showing the best and worst-case time complexity Analysis
3. Code Implementation (2 Marks):
  - Code snippet of the running merge sort program. Submit the complete program for merge sort, ensuring it includes all required features and a GitHub repository link for your code.
4. Conclusion:
  - Summarize the detailed findings, learning outcomes, and any conclusions drawn from the visualization and implementation of the sorting algorithms.

**Notes:** Assignments have to be completed individually. So, please avoid any sort of plagiarism. Please submit your assignment deliverables to Canvas.

**Deadline:** March 24 – 11:59 PM, 2024

**Contact TA:**

- Lathushan Pavalavelauthan ([lathushan.pavalavelauthan@ontariotechu.net](mailto:lathushan.pavalavelauthan@ontariotechu.net))
- Utku Soytekin ([utku.soytekin@ontariotechu.net](mailto:utku.soytekin@ontariotechu.net))