Machine Learning **F2023**

Problem Set 1

Handed Out: Sep 14^{th}, 2023 *Due: Sep* 28^{th}, 2023

In this assignment you will implement three sorting algorithms: Insertion Sort, Quicksort, and Merge Sort. You will use these to sort a variety of datasets that contain data on Pokemon and their total stats. I have given you, in total, 9 datasets that contain various subsets of the Pokemon dataset in various orders. I have given you a small dataset (only Generation 1 Pokemon), a medium dataset (only containing Generations 1-3), and a large dataset (containing Generations 1-6). Each of these also comes in a sorted, reverse sorted, and random variety.

In addition to implementing these algorithms, you will be tasked with estimating the runtime of each algorithm on each dataset by counting the number of comparisons made during execution.

Finally, you will submit a writeup in Word or PDF that summarizes your results and all code as a zip file. Submit the writeup (with attached source code) to the Canvas submission locker before 11:59pm on the due date.

Sort Pokemon! (60 points)

Here, you will implement Insertion Sort, Quicksort, and Mergesort in C++. These algorithms should be able to successfully sort all 9 of the datasets provided for you (not necessarily at the same time though!).

Details:

- You may not use a high-level function for implementing these sorts (generally, anything that simply solves the problem for you). If you have questions about whether a command is permissible, please ask and I will let you know.
- As mentioned at the start of class, all code should be written in C++.
- Please include instructions for how to compile and run your code in your writeup.
- Explain any implementation choices you had to make in the final report (such as how ties were broken).

Estimate and Graph Runtime (20 points)

In this part of the assignment, we're going to try to estimate how fast each of these algorithms is on each dataset. To do this, modify your sorting algorithms so that they can keep track of the number of comparisons made in each algorithm. Remember, every sorting algorithm must, at some point, compare one value to another to determine where it goes in the list. These are the operations I want you to keep track of.

For each algorithm on each dataset, report, visualize, or otherwise present the number of comparisons made. Does this align with what you would expect to happen given the big-O runtimes of each algorithm?

Details:

- You can choose how you want to present the data. It could be as a table containing all of the values. It could also be a graph if you'd like. You can use any tool you'd like to represent the data.
- Be sure to include a discussion of whether this behavior seems normal given what you know about how each algorithm works and the best and worst-case runtime complexities of each algorithm.
- All of this information should be included in the writeup.

Writeup (20 points)

You will include a written report with your submission detailing important details about your implementation, as well as the results of any analyses requested in the assignment. The report must be complete and clear. A few key points to remember:

- Complete: the report does not need to be long, but should include everything that was requested.
- Clear: your grammar should be correct, your graphics should be clearly labeled and easy to read.
- Concise: I sometimes print out reports to ease grading, don't make figures larger than they need to be. Graphics and text should be large enough to get the point across, but not much larger.
- Credit (partial): if you are not able to get something working, or unable to generate a particular figure, explain why in your report. If you don't explain, I can't give partial credit.

Use of Generative Machine Learning

As stated in the syllabus, you are allowed to use Generative Machine Learning models to aid in your completion of this project. If you choose to do this, you must cite your usage by providing the following information:

- Which Generative system was used? (ChatGPT, GPT4, Dall-E, etc.)
- What was the Generative AI used to create?
- The prompt used to generate the desired output.
- Any changes made to the output before submission.

If you have any questions about a specific use of a generative system, please let me know and we'll get it worked out!