Project 1

**Overview**

* In this project, students will explore and implement hybrid or adaptive sorting algorithms to improve the efficiency of the sorting process through piggybacking the existing sorting algorithms.
* Hybrid Sorting: a sorting algorithm that combines the strengths of different sorting methods to achieve improved performance.
* Adaptive Sorting: a sorting algorithm that dynamically chooses and adapt their sorting strategy based on the input data characteristics.

**Project Components**

* Research and Study: Students will start by studying various sorting algorithms.
* Algorithm Design: Students will choose hybrid sorting algorithms or adaptive sorting algorithms that combine two or more sorting techniques. Students can add their heuristics to improve the algorithm further. Or students can develop their own version algorithm from scratch.
* Implementation and Coding: Students will write code to implement the sorting algorithms in a programming language of Python.
* Testing and Benchmarking: Students will conduct rigorous testing to validate the correctness of their algorithms for given test inputs. They will also perform benchmarking experiments to compare the performance of chosen algorithms with underlying component sorting algorithms.
* Performance Analysis: Through experimental results, students will analyze and compare the performance of their sorting algorithms in terms of execution time and memory usage.

“hybrid merge sort”

“adaptive merge sort”

**What is an Adaptive and Non-Adaptive Sorting Algorithm?**

With certain sorting algorithms, if the data is already sorted, the complexity of the sorting algorithm also changes. In other words, the sorting algorithm time complexity depends on the order of the given input.

So, Adaptive sorting algorithms have their complexities change depending on the order of the elements. Also, if the data is already in order, the algorithm will not reorder the elements.

Some examples of adaptive sorting algorithms:

* + Insertion sort
  + Quick sort
  + Bubble sort

With adaptive sorting algorithms, if the data is already in order, the running time is O(n).

On the other hand, Non-Adaptive sorting algorithms do not have their complexities dependent on the order of the elements. In other words, the time complexity remains the same for an array in any order.

Some examples of non-adaptive sorting algorithms:

* + Selection sort
  + Merge sort
  + Heap sort