Week - 5 (Group 2)

Main Research Findings:

- Our algorithm, Mubble, is a hybrid sort: we split the input array into small segmentations, use Bubble Sort to sort these small pieces, and then use Merge Sort to merge the sorted subarrays. This combination takes advantage of the two algorithms. Bubble has an early stop, which is suitable for detecting local order; Merge has high efficiency and stable overall performance. By adjusting the combination logic, our hybrid algorithm is faster than both standard Bubble and Merge. We conducted multiple experiments, and hybrid sorting performed better under different input sizes.
- Compared with other hybrid sortings, Mubble is not as good as TimSort, but the main reason for the gap is that we use Bubble instead of Insertion. Bubble Sort is slower because it has more swapping, even with an early stop.

Sanity Checks & Evaluations:

- We compared several algorithms using the same dataset, including Bubble, Merge, IntroSort, and TimSort.
- Each set of tests was repeated 100 times, and the mean, max, median, and other indicators were statistically analyzed to ensure stability.
- All sorting results were checked, and the output arrays were correct, with no sorting errors found.

Partial Results & Fixes:

- We tried different thresholds and found that when the threshold changed from 32 to 10, the overall speed was significantly improved.
- To solve the problem of Mubble being slow in sorting the sorted array last week, we added a special sortedness check to reduce meaningless merges.

Future Work & Next Steps:

- If given more time, we would combine other sorting algorithms (like quicksort) to improve the performance further. We also thought about replacing Bubble with other lightweight sorting methods (such as Insertion or Selection).
- We are also planning to compare Mubble with more algorithms.
- We've started working on visualizations to prepare for next week's presentation. We present a graphical comparison of Mubble sort's performance against other algorithms, using Bubble sort–the slowest among them—as a baseline reference.