GROUP EXERCISE 1

Assignment:

Write a Code of Insertion Sort, Merge Sort and Quick Sort. Calculate the Average running time in the function of the length of Array n = 10, 50, 100,200, 500, 1000…

Show that T(n) = O (n^2) for the Insertion Sort, and T(n) = O (n lg n) for Merge Sort and Quick Sort.

Our report:

- The C++ Code (***3_sorting.cpp***) is attached with this document in .zip file

  or download link (in case of the attached code is broken): https://drive.google.com/file/d/1H7zhk52dqRD7Un4Dq0VyCJOZLpGM8cee/view?usp=sharing

  or compiling the code online: http://cpp.sh/3ucfs

- The detailed running time data file (***Sort_Alg Comparison.xlsx***) is also attached with this document in .zip file

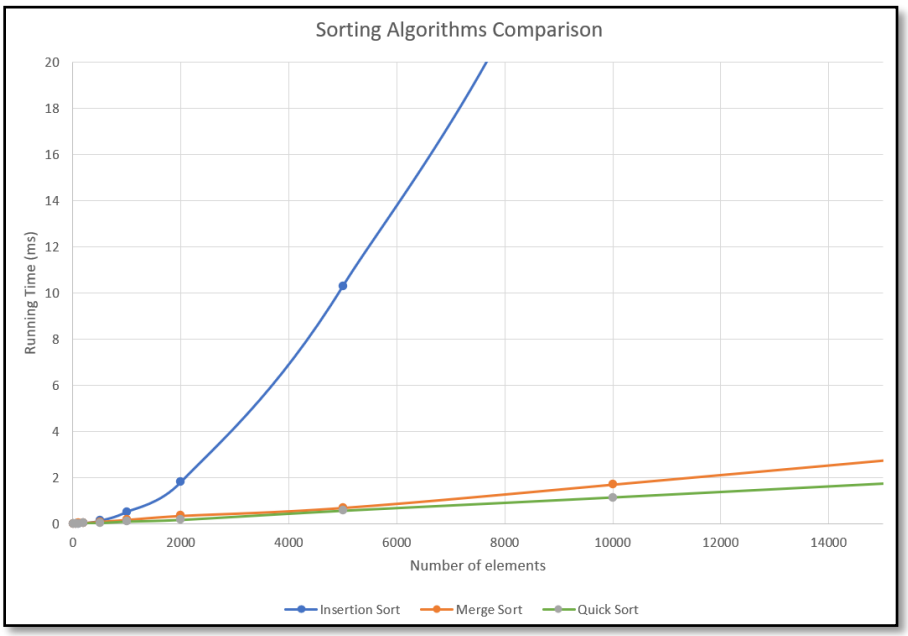
  or online link: https://docs.google.com/spreadsheets/d/1hVZ1N5sxz4Z9vAS6_Fi8QNdYfydu3-Lpeq3WMh4wSTo/edit?usp=sharing

Turning to the main part, we run 3 sorting algorithms and find out the average time for each case is in the table below:

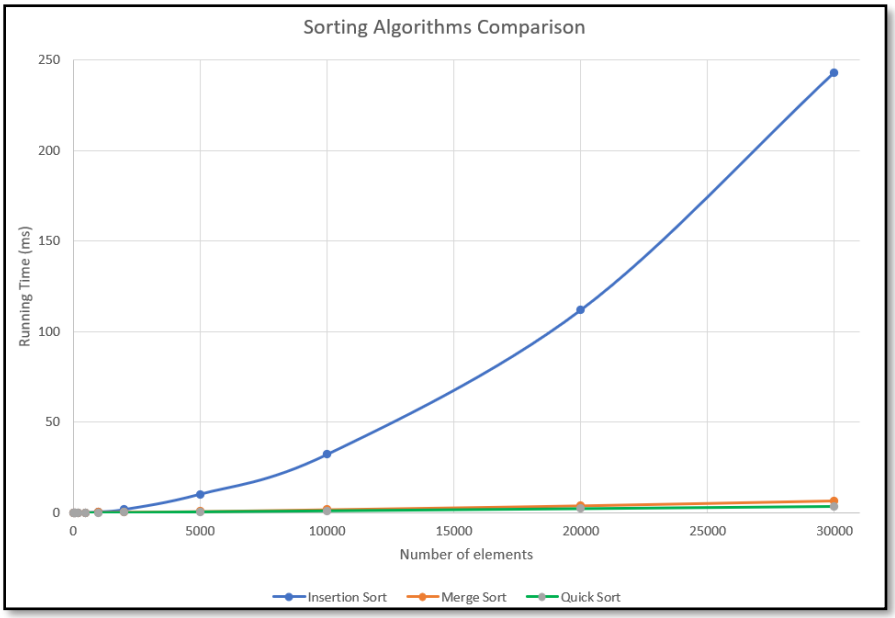
Number of elements	Insertion Sort	Merge Sort	Quick Sort
10	0	0.0004	0
50	0.002	0.006533333	0.001933333
100	0.007333333	0.0132	0.006333333
200	0.024066667	0.0282	0.013933333
500	0.128066667	0.0806	0.029266667
1000	0.517466667	0.1666	0.092666667
2000	1.807666667	0.338533333	0.1678
5000	10.3054	0.684266667	0.561133333
10000	32.19053333	1.69244	1.136066667
20000	111.9	3.857133333	2.336266667
30000	243.2189333	6.576333333	3.581466667

We use the green color to highlight the fastest time among 3 sorting algorithms, then yellow color, and finally the slowest time is filled with red color. (More details in the excel file)

Based on the data of the table above, we have the following graph:



As can be clearly seen from the graph, the average running time graph of the Insertion Sort is a curve line, which tends to increase really fast. Moreover, it has the form of the Quadratic function (polynomial function of degree 2). **Hence, the time complexity of Insertion sort is $O(n^2)$.** With regard to the Merge Sort and Quick Sort, we can notice that both of them perform a lot better than the Insertion Sort (with the same input size (number of elements), the 2 graphs are always lie under the Insertion Sort graph). Now we continue to zoom out the graph by adding more points:



It is obvious that Merge Sort and Quick Sort has the same average running time complexity due to their graph shape. Furthermore, these 2 graphs grow faster than the linear function, and much slower than Quadratic function, that means their time complexity **T(n)** is proportional to the **n lg n + a n + … (a is a constant)** function, or in other words, **the time complexity of Merge Sort and Quick Sort is O(n lg n)**.

Our Team Members

18812: Vu Hoang Tuan Anh (Team Leader)

18810: Tran Kim Hoan

17273: Truong Hoang Nam

17035: Nguyen Hoang Hai Nam

17965: Ba Nguyen Quoc Anh

17306: Tran Minh Ngoc