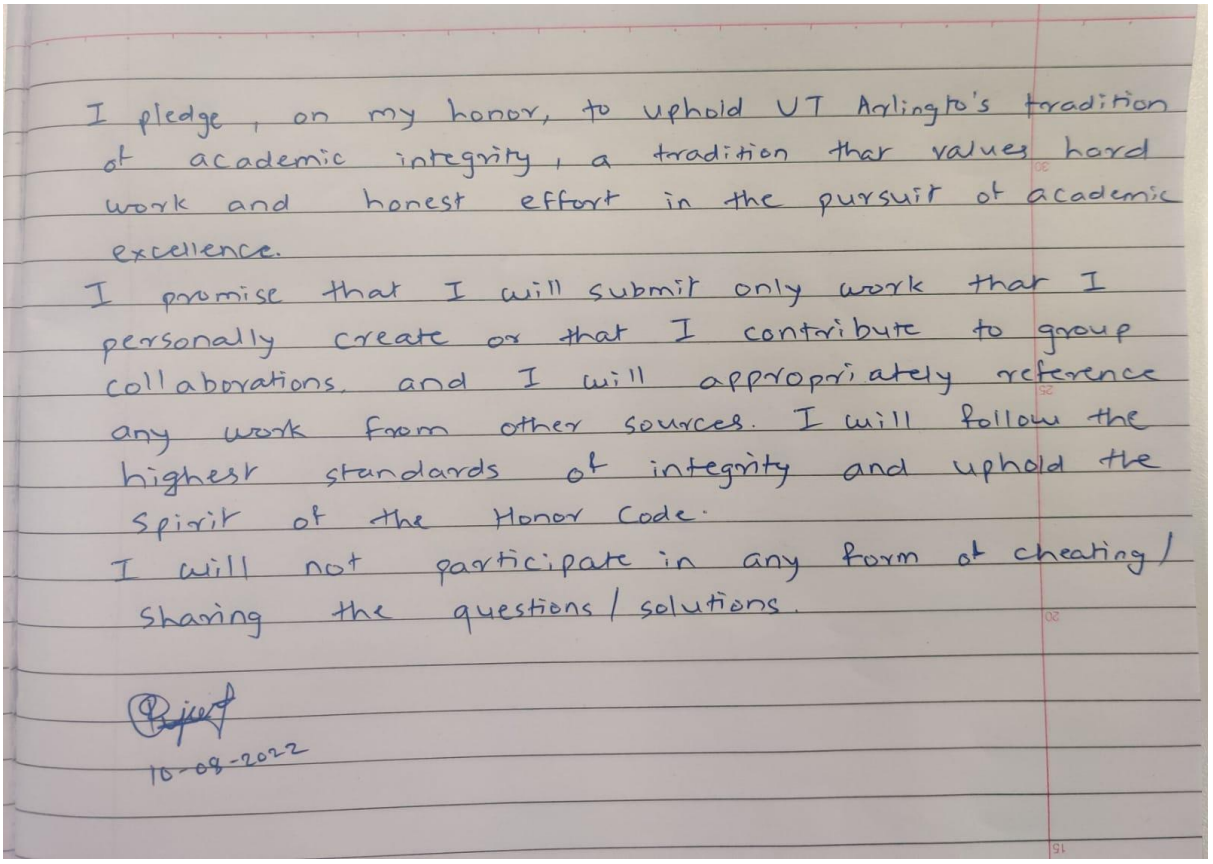
DAA Project 1 - CSE 5311

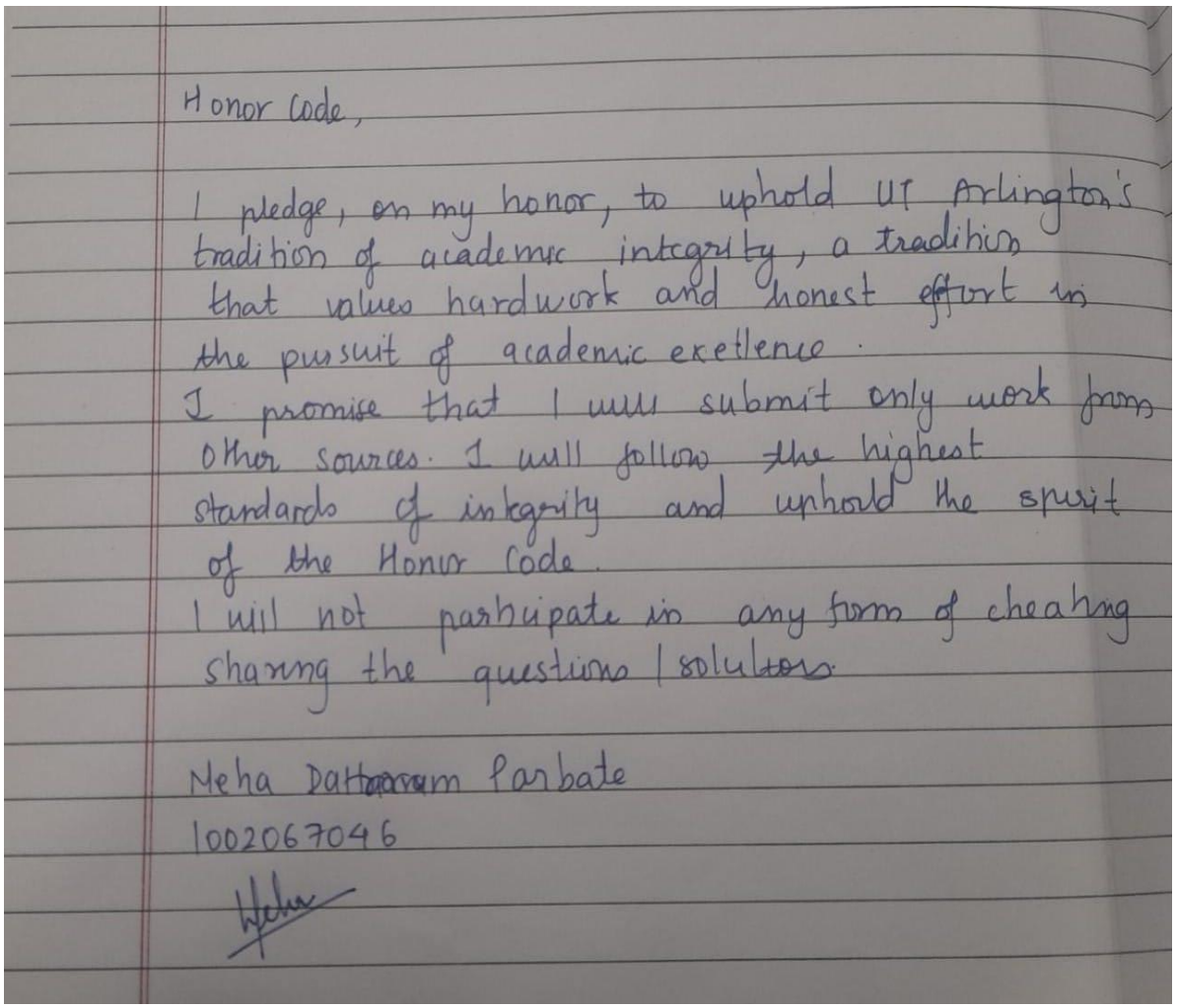
Group members:
Name: Neha Dattaram Parbate
UTA ID: 100206706

Name: Rujeet Mahesh Jahagirdar
UTA ID: 1002057347

Honor code:


I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.
I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.
I will not participate in any form of cheating/ sharing the questions/ solutions.

10-09-2022


Honor Code,

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hardwork and honest effort in the pursuit of academic exellence.
I promise that I will submit only work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.
I will not participate in any form of cheating sharing the questions / solutions.

Neha Dattaram Parbate
1002067046

Responsibility:
Neha : Implemented insertion sort, Merge sort and documented the report.
Rujeet : Implemented Merge, QuickSort and documented the report.

Time complexity of each of the algorithm:

Algorithm	Best Case	Average Case	Worst Case
Insertion Sort	O(n)	O(n^2)	O(n^2)
Merge Sort	O (n log n)	O (n log n)	O (n log n)
Quick Sort	O (n log n)	O (n log n)	O(n^2)

Experimental results:

Time in milliseconds it takes for each algorithm to sort 4 lists of data structures of 20 , 100 , 1000 and 4000 lengths .

Algorithm/Input Size	20	100	1000	4000
Insertion Sort	0.163 ms	1.265 ms	3.422 ms	2.743 ms
Merge Sort	0.056 ms	0.197 ms	0.831 ms	0.646 ms
Quick Sort	0.052 ms	0.176 ms	0.631 ms	0.503 ms

Differences between the experimental and theoretical results:

In theory the merge sort algorithm should be faster than the quick sort algorithm for a large dataset. But in a practical experiment we got quicksort to be faster than merge sort for 4000 elements array.

Compare and contrast the results between the three sorting algorithms and time taken to sort the 4 arrays. Explain anomalies if any.

From the experimental results we found out that when the array size is small insertion works faster but as the array size increases insertion sort takes more time when compared to other two merge and quicksort.

When we compared the smallest array of 20 elements, quick sort came out to be the fastest algorithm, which proves the theoretical results to be true.

Merge sort works at an almost consistent speed irrespective of the size of the array. It is faster than insertion sort for array size 100, 1000 and 4000.

We found here that insertion sort and merge sort and quicksort for 1000 elements is slightly slower than that of an array of 4000 elements.

In conclusion from the practical experiment, we can say that quicksort is the fastest algorithm when compared in a dataset for all sizes.

Sites/sources referred :

https://www.geeksforgeeks.org/
https://www.geeksforgeeks.org/python-now-function/
https://stackoverflow.com/
Professor Bhanu Jain's lectures slides