Date:

Assignment No - A3

Title :- Parallel Sorting Algorithms.

Problem Statement:-

For bubble sort and merge sort based on existing sequential algorithms, design and implement parallel algorithms utilizing all available resources.

Objectives:-

Understanding Parallel bubble and merge sort.

Outcomes:-

Understood and implemented parallel bubble and merge sort.

Software requirements:-

g++, CUDA, Google Colab, Unix OS,

Hardware requirements:-

8GB RAM, 64 bit CPU, 128 GB SSD.

Theory:-

• Bubble Sort :- there are two phases in this algorithm odd and even phases 'n' elements are sorted in 'n' phases, where n is even.
• Consider a sequence to be sorted $<a_1, a_2, \cdot \; a_n>$

The odd phase works on the odd notices are compared with their neighbours and are exchange if found out of order.
• In a similar fashion, in the even phase, the number at even phase indices are compared with their neighbours.
• The sequence is sorted after performing n phases of odd even exchanges.

Example :-

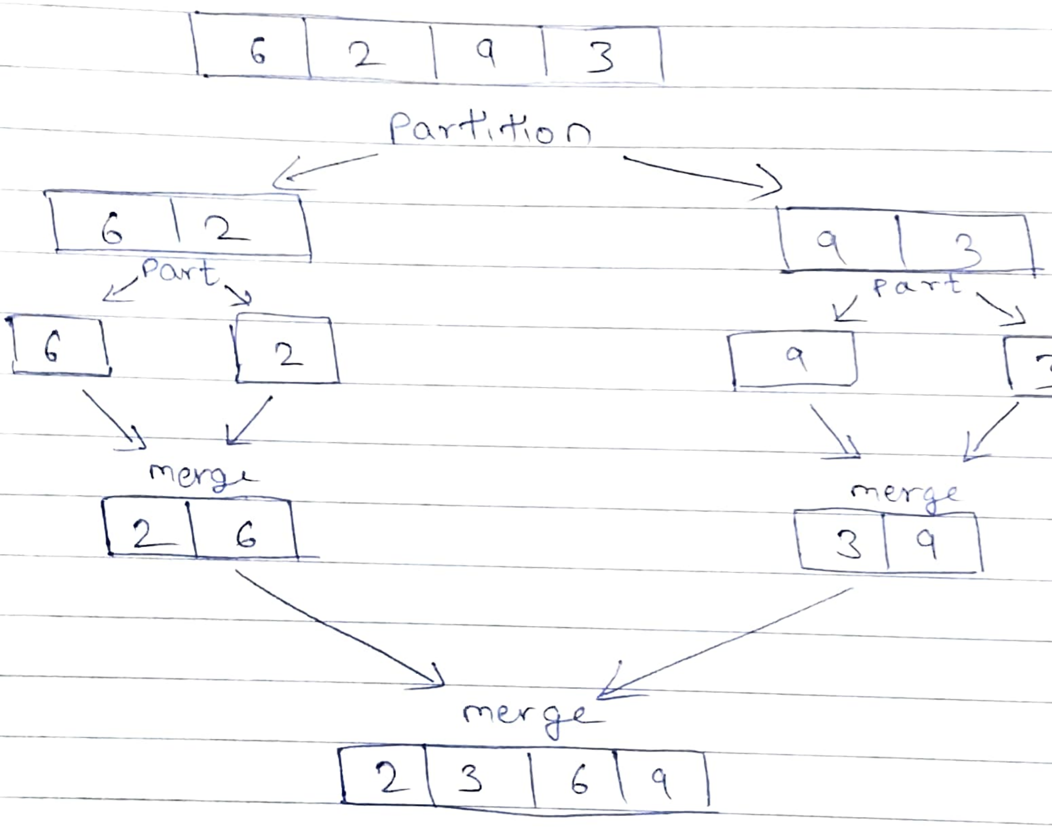
Step↓	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	4 ⟷	2	7 —	8	5 ⟷	1	3 —	6
1	4	2 —	7	8 ⟷	1	5 ⟷	3	6
2	2 —	4	7 ⟷	1	8 ⟷	3	5 —	6
3	2	4 ⟷	1	7 ⟷	3	8 ⟷	3	6
4	2 ⟷	4	4 ⟷	3	7 —	5	8 ⟷	6
5	1	2 —	3	4 —	5	7 ⟷	6	8
6	1 —	2	3 —	4	5 —	6	7 —	8
7	1	2 —	3	4 —	5	6 —	7	8

— indicates comparison
⟷ exchange.

Merge Sort first divides the unsorted list into the smallest possible sub-list, compares it with adjacent lists, them combines them accordingly.

- It implements parrallelism very well by following the divide and conquer algorithm.
- It operates in repeated partitions until no more can be achieved, followed by repeated compared - merges until the original length is achieved.

Example:-



Conclusion:-

Successfully understood and implemented Bubble and Merge Sort parallel algorithms.