Assignment No. 02

Title: Write a program to implement parallel Bubble sort and merge sort using openmp.
Use existing algorithm and measure the performance of sequential and parallel algorithm

Objective: student should be able to write a program to implement parallel bubble sort and merge sort and can measure the performance of sequential and parallel algorithm.

Theory:

Bubble sort:

Bubble sort is simple sorting algorithm that works by repeatedly swapping adjacent element if they are in the wrong order. It is called Bubble sort because the algorithm moves the larger element towards the end of the array in a manner that resembles that rising of bubbles in a liquid.

The basic algorithm of bubble sort is follows:

1. start at the begining of the array
2. compare the first two element. If first element is greater than the second element swap them.
3. move to the next pair of element repeat step 2
4. continue the process until the end the array is reached.
5. If any swaps were made in step 2-4 the process from step 1.

The time complexity of bubble sort is which makes it inefficient for large list.

Advantages and Use cases of bubble sort :

① simplicity
② Educational purposes
③ small Dataset
④ performance optimization

E.g the number are given ,

6, 3, 0, 5

step 1 :

i = 0

| 6 | 0 | 3 | 5 |

i = 1

| 0 | 6 | 3 | 5 |

i = 2

| 0 | 3 | 6 | 5 |

| 0 | 3 | 5 | 6 | sorted

step 2 :

i = 0

| 0 | 3 | 5 | 6 |

i = 1

| 0 | 3 | 5 | 6 |

| 0 | 3 | 5 | 6 | sorted

step 3

i = 0

| 0 | 3 | 5 | 6 |

| 0 | 3 | 5 | 6 | sorted Ar

merge sort :

merge sort is a sorting algorithm that a divide and conquer approach to sort array or a list of element

the algorithm works by recursively divide the input array into two halves, sorting half and then merging the sorted half produce a sorted output.

The merge sort algorithm can be broken into the following steps :

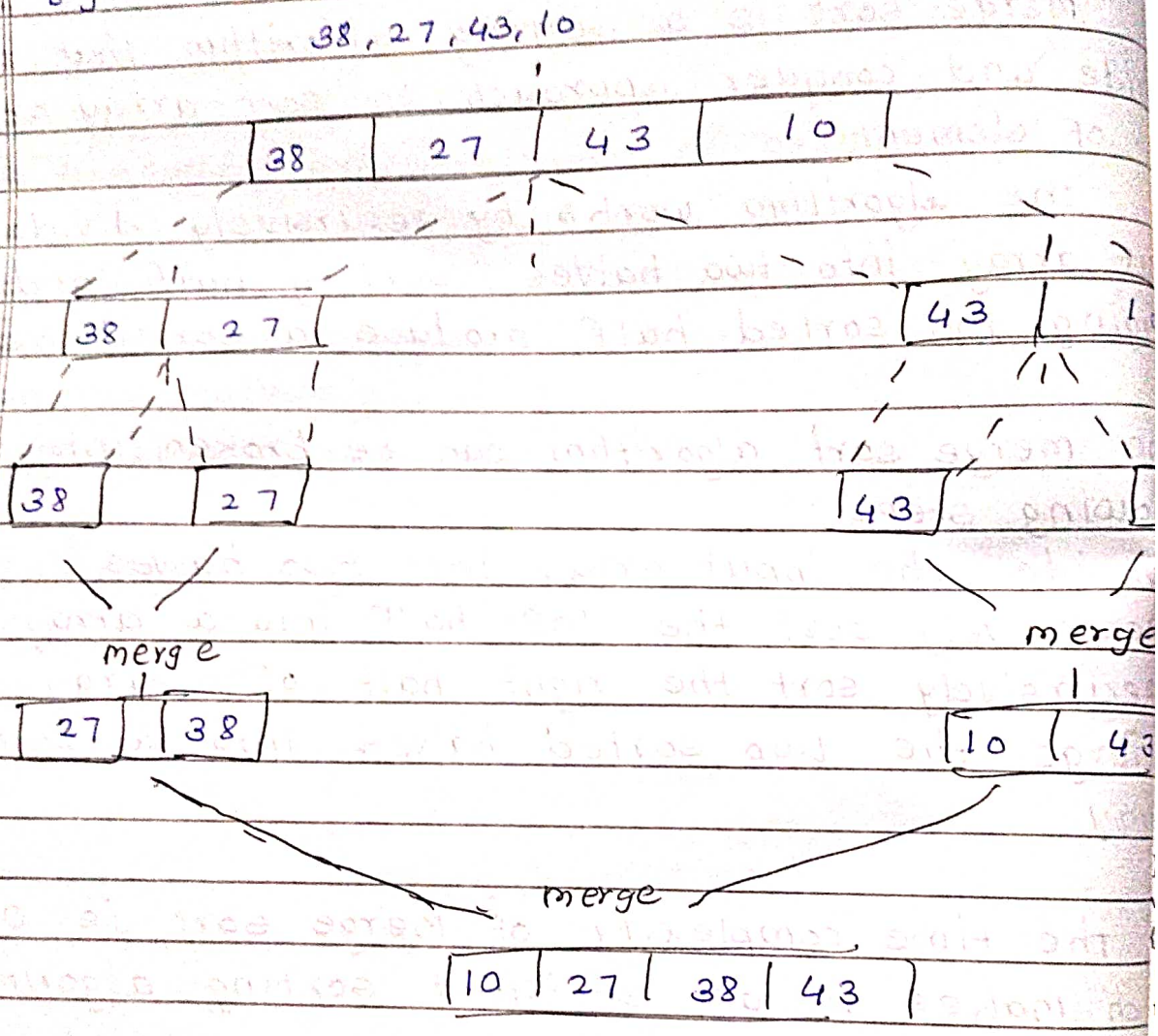
① Divide the input array into two halves
② Recursively sort the left half into a array
③ Recursively sort the right half of a array
④ merge the two sorted halves into a sorted output array.

The time complexity of merge sort is O(which makes it an efficient sorting algorithm large input arrays.

How to measure the performance of sequential parallel algorithm :

① Execution time – Used to compare the speed of algorith
② speed up – speedup is the ratio of the execution time
③ Efficiency – It is the ratio of the speed to the number of processor or used the parallel algorithm
④ scalability – It is the ability of the algorithm to maintain its performance as input size & number of processor.

E.g Numbers are given



Conclusion :

In this way, we have successfully implemented bubble sort and merge sort way using openmp also come how to measure performance of serial parallel algorithm