Roll no:- 41310

Page No.:
Date:

Assignment HPC-3

* **Title:-** Parallel sorting Algorithms.

* **Problem statement:-** for a bubble sort & merge sort based on existing sequential algorithm design and implement parallel algorithm. utilizing all resources available.

* **objective:-**
  * To understand concept of bubble sort and merge sort based on sequential algorithm.
  * To understand Concept of parallel algorithm.
  * To Compare performance by varying number of processors used and also with sequential algorithm.

* **S/w Packages and H/w apparatus:-**

Operating system :- Linux or windows.
openmp configuration.

* **Out Comes:-**

After Successfully Completing this assignment we can
* Display result for parallel bubble & merge sort.
* Analyze performance by varing number of processors used and also with sequential algorithm.

Page No.:
Date:

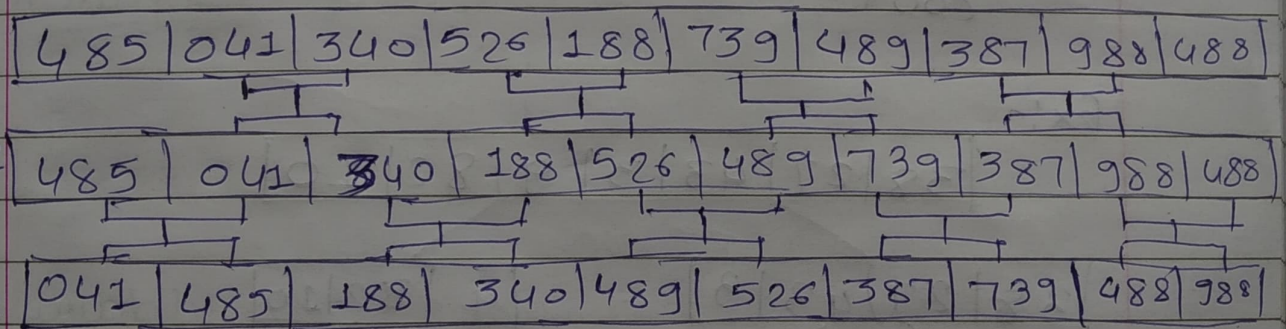
Theory:-

* Bubble sort algorithm:-

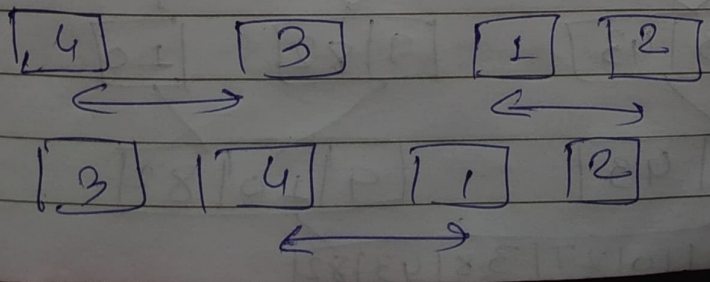
1. sequential bubble sort algorithm one of straight forward sorting method
   * Cycles through the list.
   * Compares consecutive elements and swaps them if neccesary.
   * stops when no more out of order pair.
   * slow & inefficient.
   * Average performance - O(n²)

2. Parallel bubble sort:-

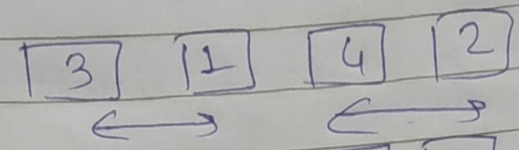
Compare all pairs in the list in parallel.



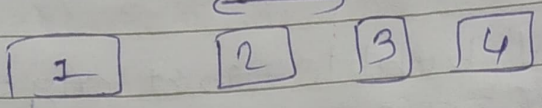
Parallel bubble sort Complexity
Do (n-1) comparison for each iteration
hence O(n)



Step 3



Step 4

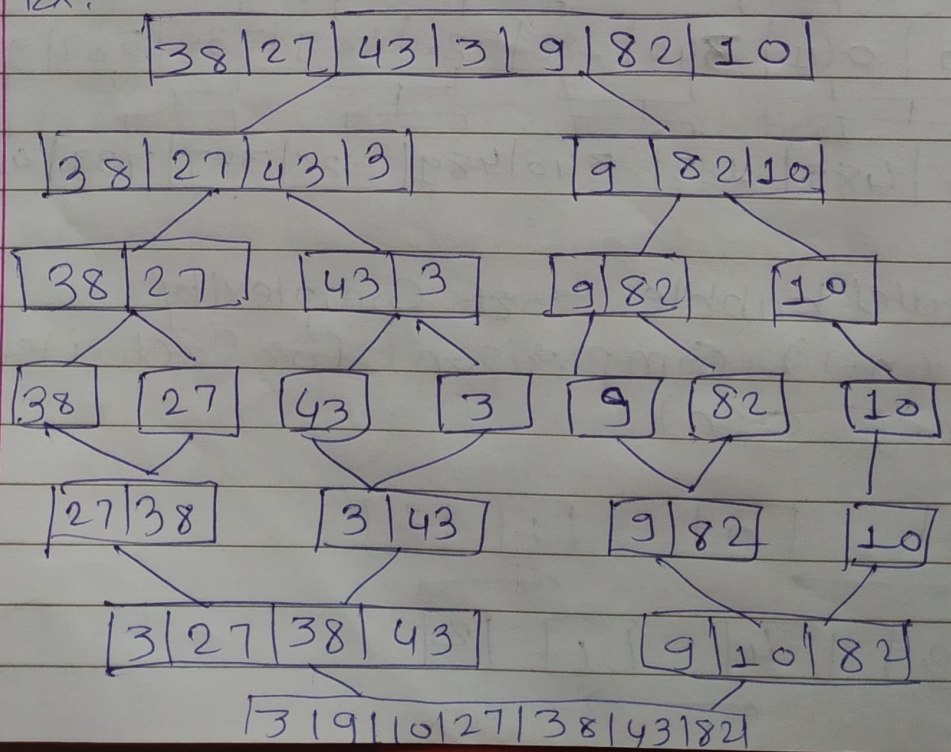


* Merge Sort Algorithm:-

Divide and Conquer approach.
* Divide problems into subproblems.
* Division usually done through recursion.
* Solution from sub-problems then Combined to give solution to original problem.

Collect sorted list onto one processor.
* Merge element as they come together
* simple tree structure.
* parallelism is limited when near the root

Ex:-

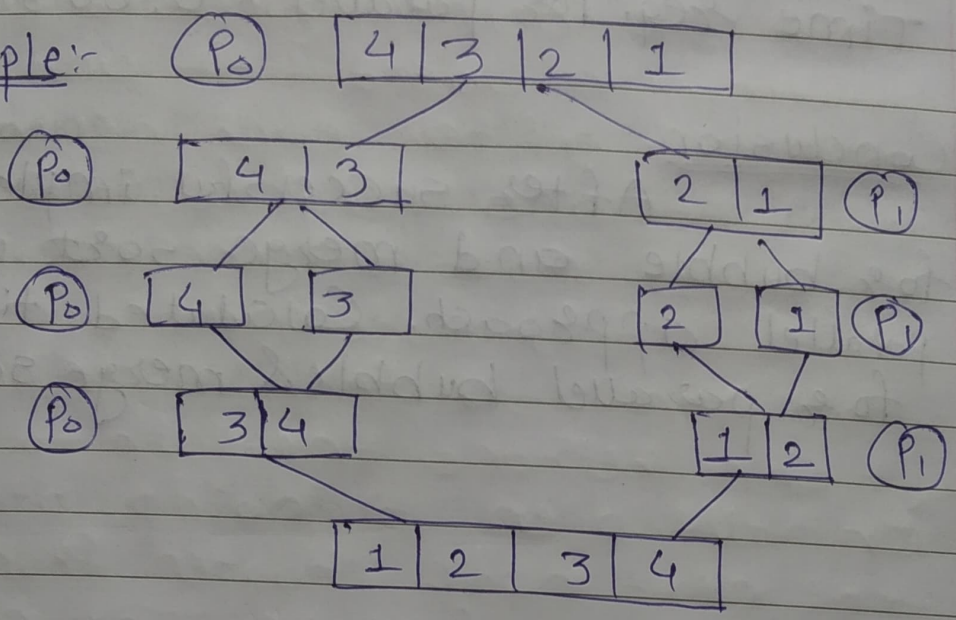


Complexity :- $T(n) = O(n \log n)$

Parallel merge Sort:-

* parallelizes problems.
* max parallelism is achieved with one processor

Example:-



Complexity:-

$T(n) = O(\log n)$

Test Cases:-

Bubble sort
Enter no. of array element 6.
Enter array elements 6 5 4 3 2 1

Sorted array: 1 2 3 4 5 6

Time required: 0.003000002.

Page No.:
Date:

Merge Sort:-

Enter no. of array element :- 6
Enter array elements 6 5 4 3 2 1.

sorted array :- 1 2 3 4 5 6.

Time req. for parallel :- 0.0039999 5.

Conclusion:-

After succesful implementation for bubble and merge sort using parallel approach. Calculated time req. for parallel bubble & merge sort.