Modern Education Society's

Wadia College of Engineering, Pune

NAME OF STUDENT:	**CLASS:**
SEMESTER/YEAR:	**ROLL NO:**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED BY:	**EXPERIMENT NO: HPC- 01**

TITLE: Parallel Sorting Algorithms

AIM: For Bubble Sort Write a program to implement Parallel Bubble Sort and Merge sort using OpenMP. Use existing algorithms and measure the performance of sequential and parallel algorithms.

OBJECTIVES:

1. To implement bubble sort and merge sort using OPENMP
2. To utilize all available resources and test on data set of sufficiently large size.
3. To compute Total cost and Efficiency.

PRE-REQUISITES:

1. Knowledge of OPENMP programming.
2. Knowledge of parallel programming.
3. Knowledge of merge sort and merge sort method.

THEORY:-

OpenMP is a set of compiler directives as well as an API for programs written in C, C++, or FORTRAN that provides support for parallel programming in shared-memory environments. OpenMP identifies parallel regions as blocks of code that may run in parallel. Application developers insert compiler directives into their code at parallel regions, and these directives instruct the OpenMP run-time library to execute the region in parallel. When OpenMP encounters the directive

#pragma omp parallel

It creates as many threads which are processing cores in the system. Thus, for a dual-core system, two threads are created, for a quad-core system, four are created; and so forth. Then all the threads simultaneously execute the parallel region. When each thread exits the parallel region, it is terminated. OpenMP provides several additional directives for running code regions in parallel, including parallelizing loops.

QUESTIONS FOR REVIEW:

1. Do the analysis of parallel bubble sort and find out its time complexity.
2. Write the difference between parallel approach bubble sort and merge sort.
3. Comment on scaling parallel merge sort.
4. How can bubble sort algorithm be parallelized?