

2023-Batch-I-Set-2

Lab 5 - CSS 311 – Parallel and Distributed Computing

OpenMP – Odd Even Transposition Sort – Monte Carlo Method

1. a) Write a C++ program for **serial bubble sort**. What is the asymptotic time complexity for this algorithm if **N** is the size of the array?
Test Case: N=10, Unsorted Array: 10 55 9 4 2 7 3 5 6 1
Expected Output: 1 2 3 4 5 6 7 9 10 55
 - b) Write C++ program for **odd-even transposition sort**, which is a variant of bubble sort. Verify the algorithm for the above test case. What is the asymptotic time complexity for this algorithm if **N** is the size of the array?
 - c) Write an OpenMP program with C++ for **Parallel odd-even transposition sort**, which is a variant of bubble sort. Verify the algorithm for the above test case. What is the asymptotic time complexity for this algorithm if **N** is the size of the array, and **P** is the total number of threads used?
-
2. **Write an OpenMP program with C++ that estimates the value of pi (π) using Monte- Carlo simulation.**
Case 1: Consider the circle with radius $r=1$ and centre (0,0) which drawn inside the square.
 - a) Write the **serial version** program to estimate the value of pi (π). Calculate the execution time by using the library function `omp_get_wtime()`.
 - b) Write the **parallel version** program to estimate the same. It includes number of threads involved and the area calculated by which thread number. Calculate the execution time by using the library function `omp_get_wtime()`.
 - c) Identify the line of statement which leads the **race condition**. Race condition occurs when the multiple threads accessing a shared variable. If it exists how will you handle this problem? Use appropriate **OpenMP directives/clauses** such as **critical**, **atomic**, **reduction** and find the solution. Calculate the execution time for critical, atomic, reduction clauses by using the library function `omp_get_wtime()`.
-