INTRODUCTION

This is the first homework for the CENG775 course. In this homework, you will implement parallel merge sort with using two MPI processes. This homework will be graded.

TASK

• Use the MPI library with C/C++ or Fortran.
• Create a distinct random array of length \( n \) (where \( n \) is an input) on each process.
• Sort this data using parallel merge sort with two processes.
• You can use any sorting algorithm in the sequential part of the parallel merge sort.
• Use efficient algorithm in merge operation. This will be also graded.
• Measure and print the wall clock time for processes 0 and 1.
• Use the mpi functions such as mpi_isend and mpi_irecv, or mpi_sendrecv.
• Can you implement the communication operation in this algorithm if you are only allowed to use mpi_send and mpi_recv?
  ◦ If yes, explain how and compare it to the implementation above (using mpi_isend and mpi_irecv, or mpi_sendrecv).
  ◦ If not, explain why.

HINTS

• Write a function for the compare and exchange operation in parallel merge sort.
• Write a function for sequentially merging two sorted lists
• Write (or use) a function for sequentially sorting an array of unsorted numbers.

TESTS

Test your algorithm using \( n = 10 \) and \( n=10000 \) on NAR cluster using 2 MPI processes. For both test cases, print the events (such as communication operations and computations), wall clock time, and sorted arrays (only for \( n=10 \)) on the screen as shown in the example below.
EXAMPLE

<table>
<thead>
<tr>
<th>Process 0</th>
<th>Process 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 8 7</td>
<td>2 5 4</td>
</tr>
</tbody>
</table>

process 0 sends its data to process 1  
process 1 sends its data to process 0  
process 1 receives the first data  
process 0 receives the second data  
process 0 merges its data  
process 1 merges its data  
sorted_data1 = [……]  
sorted_data2 = [……]  
wall clock time of process 0 = 1.0 seconds  
wall clock time of process 1 = 1.0 seconds

SUBMISSION & GRADING

This homework will be graded and submit it on cow. You submit a report which includes pseudo code of your algorithm and the results shown on the screen (except sorted data for n=10000) like in the example. This report will be submitted both hard and soft copy(.pdf). Your codes and soft copy report will be submitted on cow. **Hard copy reports should be submitted to B202.**