INTRODUCTION

This is the second homework for the CEng 775 course. In this homework, you will implement parallel merge sort with using multiple MPI processes.

TASK

- Use the MPI library with C/C++ or Fortran.
- Create a reverse order sorted array of length $n$ whose values are from $n$ to 1. Split this data to the processes like below:
  $$a_i = [n - (i * n/p), n - (i * n/p) - 1, \ldots, n - ((i + 1) * n/p) + 1]$$
  where $a_i$ is the rank of $i$th process and $p$ is the number of processes.
- Sort this data using parallel merge sort with $p$ mpi_processes.
- You can use any sorting algorithm in the sequential part of the parallel merge sort.
- Use compare and exchange function which is done in homework1.
- Draw the graphics which are defined in below part and make examinations about it.
- Print sorted arrays (only for $n = 80$) on the screen like Homework 1 (only sorted arrays not the others). **This part also graded.**

HINT:

Try to understand the algorithm which is given in the following links:
http://www.mcs.anl.gov/~itf/dbpp/text/node124.html#algbutalg

TESTS

Test your algorithm using $p = 1$, $p = 2$, $p = 4$ and $p = 8$ processors separately. Take $n = 80, n = 800000$ and $n = 8000000$ on each test. Show the scalability of tests in the same graphic by drawing the speed improvement which is:

- Number of processors versus the speed improvement for $n = 800000$ and $n = 8000000$.

**Hint:** Run the algorithm for $p = 1$ and measure $T_s$. And then calculate $\frac{T_2}{T_s}$, $\frac{T_4}{T_s}$, and $\frac{T_8}{T_s}$ for the others. Draw the graphics by using these computations.
SUBMISSION & GRADING

You submit your codes and a report on cow. This report have to include pseudo code of your algorithm, your graphics and comments related to the graphics.