Homework 1: Counting Sort (5 points)

Parallelise the counting sort algorithm using MPI as follows:

1. Initialise the array with random numbers;
2. Scatter the array to all processes;
3. Count the array elements in parallel on each process;
4. Gather the locally counted elements and use the reduction function to add them if necessary;
5. Construct the sorted array.

Homework 2: Bucket Sort (5 points)

Parallelise the Bucket Sort algorithm using MPI as follows:

1. Initialise the array with random numbers;
2. Construct the buckets and scatter them to all processes;
3. Sort the buckets in parallel on all processes using quicksort;
4. Gather and merge the sorted buckets.

Homework 3: Selection Sort (5 points)

Parallelise the Selection Sort algorithm using MPI as follows:

1. Initialise the array A[N] with random numbers;
2. Sequentially iterate over the array elements with an index I;
3. Scatter the elements from the array A[I:N] to all the processes;
4. Find the local minimum in each process in parallel;
5. Use the reduction function to find the global minimum and exchange it with A[I].

Homework 4: Performance Analysis (5 · 3 points)

1. Choose a large problem size (i.e. number of array elements) and execute each algorithm for 1, 2, 4, 8 and 16 parallel processes;
2. Compute the speedup and efficiency for each algorithm;
3. Repeat the steps 1 and 2 until you find a problem size that gives you good speedup and efficiency results.