CSci 5451, S'20	Homework # 1	Due Date: 02-12-20

1. Recall that scaled speedup is defined by selecting the problem so that work is increased linearly with the number of processing elements; that is, if a base problem of a certain size involves work W on a single processing element, then on p processors, the size is selected so that the work is pW and then the scaled speed-up is

$$S_{\text{Scaled}} = \frac{pW}{T_p(pW, p)}.$$

Consider the problem of adding n numbers on p processing elements. Assume that it takes ten time units to communicate a number between two processing elements, and that it takes one unit of time to add two numbers.
(a) Plot the standard speedup curve for the case when we are adding $n = 256$ numbers on $p = 1, 4, 16, 64$, and 256 processors. (b) On the same figure plot the scaled speedup curve. Hint: The parallel run time is $(n/p - 1) + 11 \log p$.

2. (Continuation of previous exercise) Using the expression for the standard speed-up from the previous question (same value of $n = 256$) obtain the efficiency for $p = 1, 2, 3, \cdots, 10$. How many processors can be used if we want the efficiency to be no less than 50%.

3. Assume that you have two matrices of size $n \times n$ and p processors, configured on a $\sqrt{p} \times \sqrt{p}$ torus. Obtain a timing model for the Cannon algorithm. Assume that communicating m data items to any nearest neighbor costs $t_s + m * t_w$ time units [only one channel (east, west, north, or south) can be used at a time] and that one *flop* costs one time unit. Next perform a scaled speed-up analysis where the scaling is based on memory rather than time. So when p increases, n is increases as $n = \sqrt{p} n_0$, where n_0 is the initial size. On the same figure plot the standard and scaled speed-up for $n_0 = 32$ and when $\sqrt{p} = 2^k$ and $k = 0, 1, 2, 3, 4$ (i.e., $p = 1, 4, 16, 64, 256$). Assume $t_s = 10, t_w = 1$.

4. Show the progression of the (a) odd-even Mergesort algorithm, and the (b) bitonic sort algorithm, when sorting the array

$$3,\ 10,\ 1,\ 11,\ 30,\ 18,\ 8,\ 20.$$

[Hint: follow similar detail of process as for the examples in the notes.]

5. Show a sorting network based on bitonic sort to sort a sequence of 4 numbers. Take the numbers 9, 2, 7, 4 as an example and show how the sorting network processes them. Then show a sorting network based on odd-even Mergesort to sort a sequence of 4 numbers. Use the same example as above to illustrate. Finally, extend these two pictures for the situation of $n = 8$ entries. Show the progress of both algorithms on the resulting sorting networks for the sequence 9, 2, 7, 4, 10, 1, 3, 5. How many comparators are required to build a bitonic sorting network which sorts 8 numbers? How many will be required for the Odd-Even mergesort?

6. Find a recurrence relation that gives the total number of operations needed to perform an Odd-Even Merge operation (sorting two arrays of length $n/2$ each into an array of length n by the OEmerge algorithm). Solve this recurrence relation. [Hint: 1) we only count comparisons. 2) it may help to write $n = 2^k$; 3) use induction to prove the result]. Use this to find the total time needed to sort an array of length n by the Odd-Even MergeSort algorithm.