Exercises: Partitioning

1. Develop pseudo-code for parallel prefix.

2. Write out the pseudo-code the iterative tree pattern used in bottom-up mergesort using bit shift operators.

3. Write out pseudo-code for testing if the set of numbers distributed across the cluster are sorted correctly, which is defined below.
   
   At the end of the sorting, process $i$ has elements that are sorted and any element on process $i$ is greater than or equal to any element on process $j$, where $j < i$ and any element on process $i$ is less than or equal to any element on process $k$, where $k > i$.

4. A sequential formulation of quicksort is given below:

   ```
   QUICKSORT(A,p,r)
   1. if (p < r)
   2. then  q ← PARTITION(A,p,r)
   3. QUICKSORT(A,p,q)
   4. QUICKSORT(A,q+1,r)
   ```

   Describe a parallel implementation of quicksort on a Beowulf cluster. Parallelize the partitioning step across the cluster. You can assume that each of the $p$ processors initially holds $n/p$ numbers. At the end of sorting, the roughly $n/p$ numbers at processor $p_i$ are smaller than all of the $n/p$ numbers at processor $p_j$ for $i < j$. Note that the total number of elements that end up at each process will depend upon the distribution of the numbers.