Bitonic Sort

Bitonic sequence:
- A non-decreasing sequence followed by a non-increasing sequence
- A non-increasing sequence followed by a non-decreasing sequence
- Or a cyclic shift of one of the above
Bitonic Merge:
for (i = 0; i < len; i++)
    if (left_seq[i] > right_seq[i])
        swap(&left_seq[i],&right_seq[i]);

Key Property: If two sorted sequences are merged, then the result is two bitonic sequences such that all values in the lower bitonic sequence are lower than all values in the upper bitonic sequence.
2-Proc Algorithm

- Proc 0 sorts the data in ascending order
- Proc 1 sorts the data in descending order
- They perform a bitonic merge
- Proc 0 sorts the data in ascending order
- Proc 1 sorts the data in ascending order
local_sort(my_array,bit(rank,0));
for (depth = 1; depth <= log2(P); depth++){
    for (p = depth-1; p >= 0; p--) {
        prank = flipbit(rank,p); // rank of partner
        parray = get_partner_data(prank);
        if (bit(rank,d) == 0)
            merge_up(my_array,rank,parray,prank);
        else
            merge_down(my_array,rank,parray,prank);
    } // loop over subphases
    local_sort(my_array,bit(rank,depth));
} // loop over depths of “recursive” algorithm
Additional Functions

\textbf{bit}(rank,\text{bitnum}) : returns \text{bitnum}th bit from rank, e.g.
returns 0 for rank 0, 2, 4, etc. and 1 for rank 1, 3, etc.

// Make my\_array have the appropriate values.
// The process with the higher rank keeps the
// higher values.
merge\_up(my\_array, rank, parray, prank, ret) {
  for i = 0; i < length\_per\_proc; i++ {
    if (rank < prank) // I am the lower rank
      if (my\_array[i] >= parray[i])
        my\_array[i] = parray[i];
    else // I am the higher rank
      if (my\_array[i] < parray[i])
        my\_array[i] = parray[i];
  }
}
Links

Website with animations and a great explanation, but they don’t write the algorithm the same way we do


Website about C++ code that does write the algorithm like we do

http://www.cs.uoregon.edu/research/paraducks/papers/psc94.d/node2.html