Like Chen, Huang and Langston describe a linear time in-place merging procedure that can then be used to implement a full merge sort algorithm.   
The merge is accomplished by moving the √n largest elements of the list to the front and using them as internal buffer.   
The remainder of each sublist is then divided into blocks of size √n. In case this is not possible without at least one undersized block we must do some preparatory work first.  
We omit the details here and assume the list is ready for the main algorithm to commence.  
First, the blocks need to be sorted by their rightmost elements.   
Selection sort is a good choice for this, as it is very easy to implement and performs only O(n) swaps.  
  
Afterwards we need to determine the next two series of elements to be merged.  
The first series consists of the first block after the buffer and all following blocks until one block’s rightmost element is larger than the next block’s leftmost element. The second series consists only of the latter block. These two series are merged, beginning at the leftmost buffer position and breaking ties in favour of the first series, until the first series is exhausted. Therefore, the second series will still contain at least one element and the buffer elements will be back in one piece, adjacent to the left of the second series’ remainder.   
This process, starting with finding the two series, is now repeated with the first “block” of the new first series being the remainder of the last second series, until there is no second series to be found.  
  
Then we just need to perform a cyclic shift of the remaining elements and the buffer, leaving the list sorted except for its √n rightmost elements, consisting only of the buffer elements. Since these are the largest elements of the list, we just need to sort them in order to finish sorting the whole list.  
The paper of Huang and Langston does not provide a specific suggestion as to how the buffer should be sorted. For the purpose of using solely the algorithm in the paper, we had the buffer sorted by a recursive call to our merge sort implemented with the described merge of Huang and Langston.

We will give a quick outline of our implementation’s basic procedures.  
Note that n is always referring to the size of the list, “merge” is initially called with.

mergesort: Divides the given list into two sublists of equal size. The first sublist is always sorted with another merge sort algorithm, using the second one as extra space, while the second sublist is handled by a recursive call to “mergesort”. The recursion stops when the list contains 50 or less elements and the remaining list is sorted by insertion sort. Both sublists are then merged using the “merge”-procedure.

merge: Determines the sqrt(n) largest elements, that will constitute the buffer and whether the remaining sublists meet the size requirements (integral multiples of sqrt(n)). If the requirements are met, the basic merge procedure, as described above, can commence. The buffer will then be extracted, using 2 cyclic shifts and “basic\_inplace\_merge” is called. Otherwise some undersized blocks must be extracted alongside the buffer, so we can again use “basic\_inplace\_merge” on the remaining list (and merge one of the undersized blocks afterwards).

basic\_inplace\_merge: This is where the actual merging begins. As explained above, we now regard our list, excluding the buffer, as a list of blocks, each of size sqrt(n). The blocks need first be sorted by their rightmost element. This is done by “sort\_blocks”, which is essentially an implementation of selection sort. The following process of finding new series and merging them is accomplished by alternating calls to “merge\_with\_buffer” and “find\_series”, the first one obviously performing the merge and returning the start of the new first series. The latter one simply returns the start of the new second series or, if no such series can be found, the end-iterator, that serves as condition to terminate the loop. Afterwards, the buffer is shifted to its final place and sorted. For the purpose of testing only the merge of Huang and Langston, this is done by using “mergesort”.

merge\_with\_buffer: Merges the first and second series, with the resulting list beginning at the first buffer position, until the first series’ elements are exhausted. Ties are broken in favour of the first series. As a result, the buffer will be in one place after the procedure, adjacent to the right of the sorted list and followed by at least one unmerged element of the second series.