WE CLAIM:

1. A method comprising:

dividing, by a computing system, data items in a spreadsheet table into a plurality of blocks, each of the plurality of blocks having a size defined by a total number of spreadsheet table rows, the size for at least one of the plurality of blocks being determined by a value associated with the total number of spreadsheet table rows falling between a first threshold number of rows and a second threshold number of rows;

using multiple threads to sort the data items in the blocks;

after sorting the data items in the blocks, using multiple threads to merge the blocks into a final block, the final block containing each of the data items in the spreadsheet table; and

displaying a sorted version of the spreadsheet table in which the data items in the spreadsheet table have the same order as an order of data items in the final block, the data items comprising hidden and visible data items, the hidden and visible data items comprising hidden and visible rows, the sorted version of the spreadsheet table comprising a sort of the hidden and visible rows, the hidden rows comprising rows that are in the spreadsheet table, the rows being hidden from view to simplify an appearance of the spreadsheet table with respect to a spreadsheet application user.

~~2. The method of claim 1, further comprising:~~

~~determining an appropriate block size based on a number of data items in the spreadsheet table; and~~

~~wherein the data items in the spreadsheet table are divided into the plurality of blocks such that none of the blocks contains more data items than the appropriate block size and only one of the blocks is allowed to contain fewer data items than the appropriate block size.~~

~~3. The method of claim 2, wherein determining the appropriate block size comprises determining that the appropriate block size is a given size when the total number of data items in the spreadsheet table is greater than one threshold and less than or equal to another threshold.~~

24. The method of claim 1, wherein the data items in the final block are ~~properly~~ ordered for multiple sort-by columns.

35. The method of claim 1, further comprising:

determining whether a total number of data items in the spreadsheet table exceeds a ~~lower~~ limit; and

using a single thread to sort the data items in the spreadsheet table when the total number of data items in the spreadsheet table does not exceed the ~~lower~~ limit.

4~~6~~. The method of claim 1,

wherein the method further comprises: prior to sorting the data items in the blocks, determining ~~an appropriate number of~~ block sorting threads for the spreadsheet table;

wherein using multiple threads to sort the data items in the blocks comprises using the ~~appropriate number of~~ block sorting threads to sort the data items in the blocks.~~;~~

~~wherein the appropriate number of block sorting threads is equal to a number of the blocks when the number of the blocks is less than or equal to a number of processing units in a processing system,~~

~~wherein the appropriate number of block sorting threads is equal to the number of processing units in the processing system when the number of blocks is greater than or equal to the number of processing units in the processing system.~~

5. The method of claim 4, further comprising: presenting a user interface that allows an administrative user to set the block sorting threads.

6. The method of claim 1, wherein using multiple threads to sort the data items in the blocks comprises using a merge sort algorithm to sort the data items in the blocks.

7. The method of claim 1, wherein using multiple threads to merge the blocks into the final block comprises:

~~waking a min merge thread that progressively inserts smallest data items in the sorted blocks into the final block; and~~

waking a ~~max merge~~ thread that progressively inserts one or more of the ~~largest~~ data items in the sorted blocks into the final block.

8. The method of claim 7,

~~wherein the min merge thread uses a first red-black tree to identify the smallest data items in the sorted blocks; and~~

wherein the ~~max merge~~ thread uses a ~~second~~ red-black tree to identify the one or more of the ~~largest~~ data items in the sorted blocks.

9. The method of claim 1, wherein displaying the sorted version of the spreadsheet table comprises: sending result data to a client device via a network, the client device configured to process the result data for presentation of the sorted version of the spreadsheet table to a user.

10~~9~~. The method of claim 1~~7~~,

~~wherein the multiple threads used to sort the data items in the blocks operate concurrently.; and~~

~~wherein the min merge thread and the max merge thread operate concurrently.~~

11~~10~~. A computing system comprising:

a processing system that comprises a plurality of processing units, each of the plurality of processing units comprising a physical integrated circuit; and

a data storage system that stores computer-readable instructions that, when executed by one or more of the plurality of processing units, cause the computing system to:

divide the data items in a spreadsheet table into a plurality of blocks, each of the plurality of blocks having a size defined by a total number of spreadsheet table rows, the size for at least one of the plurality of blocks being determined by a value associated with the total number of spreadsheet table rows falling between a first threshold number of rows and a second threshold number of rows;

use multiple threads to sort the data items in the blocks based on a relevant property of cells in a sort-by line of the spreadsheet table;

use multiple threads to merge the blocks into a final block, the final block containing each of the data items in the spreadsheet table; and

display a sorted version of the spreadsheet table in which the data items in the spreadsheet table have the same order as an order of ~~the~~ data items in the final block, the data items comprising hidden and visible data items, the hidden and visible data items comprising hidden and visible rows, the sorted version of the spreadsheet table comprising a sort of the hidden and visible rows, the hidden rows comprising rows that are in the spreadsheet table, the rows being hidden from view to simplify an appearance of the spreadsheet table with respect to a spreadsheet application user.

12~~11~~. The computing system of claim 11~~10~~, wherein the computer-readable instructions, when executed by one or more of the processing units, cause the computing system to determine a ~~an appropriate~~ block size based on the ~~number of~~ data items in the spreadsheet table~~, wherein none of the blocks has more data items than the appropriate block size~~.

13~~12~~. The computing system of claim 11~~10~~, wherein the computer-readable instructions, when executed by one or more of the plurality of processing units, further cause the computing system to determine ~~an appropriate number of~~ block sorting threads,

~~wherein the appropriate number of block sorting threads is equal to a number of the blocks when the number of the blocks is less than or equal to a number of the processing units in the processing system,~~

~~wherein the appropriate number of block sorting threads is equal to the number of processing units in the processing system when the number of blocks is greater than or equal to the number of processing units in the processing system, and~~

wherein the computing system uses the ~~appropriate number of~~ block sorting threads to sort the data items in the blocks.

14~~13~~. The computing system of claim 11~~10~~, wherein to use multiple threads to merge the blocks into the final block, the computer-readable instructions, when executed by one or more of the processing units, cause the computing system to:

~~wake a min merge thread that progressively inserts smallest data items in the sorted blocks into the final block; and~~

wake a ~~max merge~~ thread that progressively inserts one or more of the ~~largest~~ data items in the sorted blocks into the final block.

~~14. The computing system of claim 13,~~

~~wherein the min merge thread and the max merge thread operate concurrently; and~~

~~wherein the multiple threads used to sort the data items in the blocks operate concurrently.~~

15. The method of claim 14,

wherein the merge thread uses a red-black tree to identify the one or more of the data items in the sorted blocks.

16~~15~~. A computer-readable data storage medium that stores computer-readable instructions that, when executed by one or more processing units comprising a physical integrated circuit in a processing system of a computing system, will cause the computing system to:

determine whether ~~a total number of~~ data items in a spreadsheet table exceeds a ~~lower~~ limit;

when the ~~total number of~~ data items in the spreadsheet table does not exceed the ~~lower~~ limit, use a single thread to sort the data items in the spreadsheet table;

when the ~~total number of~~ data items in the spreadsheet table ~~is equal to or~~ exceeds the ~~lower~~ limit:

determine a first ~~that an appropriate~~ block size based on a ~~is a first size when the~~ total number of ~~data items in the~~ spreadsheet table rows falling between ~~is greater than~~ a first threshold and ~~less than or equal to~~ a second threshold associated with the spreadsheet table;

determine a second ~~that the appropriate~~ block size based on ~~is a second size when the total number of data items in the spreadsheet table is greater than~~ the second threshold, the second block size being larger than the first block size;

divide the data items in the spreadsheet table into a plurality of blocks;~~,~~ ~~none of the blocks containing more data items than the appropriate block size and only one of the blocks being allowed to contain fewer data items than the appropriate block size;~~

determine ~~an appropriate number of~~ block sorting threads for the spreadsheet table;~~,~~

~~wherein the appropriate number of block sorting threads is equal to a number of the blocks when the number of the blocks is less than or equal to a number of the processing units in the processing system,~~

~~wherein the appropriate number of block sorting threads is equal to the number of processing units in the processing system when the number of blocks is greater than or equal to the number of processing units in the processing system,~~

use a plurality of the block sorting threads to sort the data items in the blocks~~, the block sorting threads being equal in number to the appropriate number of block sorting threads~~; and

after the plurality of the block sorting threads have sorted the data items in each of the blocks, use a ~~min merge~~ thread ~~and a max merge thread~~ to merge the data items in the blocks into a final block, the final block containing each of the data items in the spreadsheet table, the ~~min merge~~ thread progressively inserting one or more of the ~~smallest~~ data items in the sorted blocks into the final block~~, the max merge thread progressively inserting largest data items in the sorted blocks into the final block~~; and

display a sorted version of the spreadsheet table in which the data items in the spreadsheet table have the same order as an order of data items in the final block, the data items comprising hidden and visible data items, the hidden and visible data items comprising hidden and visible rows, the sorted version of the spreadsheet table comprising

a sort of the hidden and visible rows, the hidden rows comprising rows that are in the spreadsheet table, the rows being hidden from view to simplify an appearance of the spreadsheet table with respect to a spreadsheet application user.