

Put on the SAS® Sorting Hat and Discover Which Sort is Best for You!

Louise Hadden, Abt Associates Inc
Charu Shankar, SAS® Canada

ABSTRACT



Sorting in SAS® is an expensive process in terms of both time and resources consumed. In this session, prepare to explore some of the common and lesser known sorts that SAS provides. Become like the sorting hat in Harry Potter! Instead of waiting with baited breath for your team (or data) to be sorted, get the inside scoop and learn about the dynamic processes that go on behind the scenes during sorting that will enable you to pick the very best sort for your circumstances. Learn about some fantastical, magical SAS sorting teams: bubble sort, quick, threaded and serpentine. Behold the effervescent bubble sort! In a hurry? Take a look at the quick sort. Looking for superior efficiency? Consider the threaded sort. See how the hissing serpentine sort in SAS, like the slithering serpent Nagini sliding surreptitiously through walls, can come in handy! Which sort will you choose – or which sort will choose you?

INTRODUCTION



Sorting of data is one of the most ubiquitous needs in SAS programming, whether the requirement is the need to vertically sort data, or sort arrays (horizontal sort). Vertical sorts arrange data across observations, by one or more variables. Horizontal data sorting sorts data across variables – for example, if a data set contains information for ten hospital visits, perhaps these visits need to be sorted by visit date or visit type. As with vertical sorting, horizontal sorting may need to be performed using one or more variables. Additionally, PROC SORT options can provide a SAS practitioner with the means to customize the order of variables and records within a data set and methods to enhance efficiency: for example, the TAGSORT and collating options such as ASCII or FINNISH. It is also possible to sort data via procedures other than PROC SORT, such as PROC REPORT, PROCs with CLASS statements, and PROC SQL.

SAS does not provide a function or call routine for the sorting of arrays (horizontal sorts). Horizontal sorts can be most helpful and can illuminate trends in data. It is left to the SAS practitioner to find a macro routine to perform a horizontal sort, or write one themselves. Techniques used to perform horizontal sorts include do-loops, arrays, transpose, etc. Two existing routines to perform horizontal sorts are the bubble and quick sorts, both described in this paper.

Vertical sorts are most common in SAS, and can be accomplished using a variety of techniques, including PROC SORT (and options), PROC SQL (order by) and implicit sorting (sorting by means of procedures other than PROC SORT (and options.)) A sort option (THREADED) and an example of implicit sorting (PROC SURVEYSELECT) will be discussed in this paper. Some lesser known but useful options will also be discussed in this paper.

THE SORTING HAT

In order to determine the best sorting technique in a given situation, SAS practitioners must carefully evaluate both the current data structure of the table to be sorted and the desired output. Key questions that must be answered are whether or not the data elements to be sorted exist in a horizontal or vertical structure. Subordinate questions include: whether the data set to be sorted is small, medium-sized or large? Does the data set to be sorted have few or many variables? How much disk space / memory is available? Is time of the essence? How many key variables are in the data set to be sorted? What purpose is the data set to be sorted destined for? What language are the data to be sorted? Can the data set to be sorted be manipulated or pre-processed to facilitate sorting?

TRANSFIGURATION CLASS

There are many ways to sort a file in SAS, and many excellent papers discussing various methodologies (please see the reference section below). This presentation reveals four more esoteric SAS sorting techniques, with the hopes that readers will follow their curiosity to learn more about each of these fantastical sorts! Two horizontal sort options will be explicated; the bubble and quicksort algorithms. Vertical sort options include a sort with the THREADS option; and an implicit technique within PROC SURVEYSELECT, a serpentine sort. For smaller data sets with fewer variables which require a horizontal sort, the bubble sort is the answer. For larger data sets which require a horizontal sort, the quick sort is the answer. For greater efficiency and processing speed in a vertical sort, a threaded sort is the answer. The CONTROL statement in PROC SURVEYSELECT has a default hierarchical (serpentine) sort for complex survey sample designs. We will discuss these four sort techniques below.

BUBBLE SORT

The bubble sort is the most frequently used horizontal sort, and is appropriate for data sets with items to be sorted arrayed horizontally, for example type1, cost1, type2, cost2, etc.. The name comes from the fact that smaller values continuously move to the top, bubbling up, as the process iterates. Array processing is a key element of this sorting technique. Arrays are set up to hold variables or values, and then do loops and conditional processing are utilized to compare adjacent variables or values repeatedly to reorder the variables or values. This method takes a given array of values or variables representing values and iterates through the array comparing each adjacent value and swapping until there are no more swaps and the correct order is confirmed.

Usage:

```
Data b;
  Set a;
  Array c(3) diastbp1-diastbp3;
  Do i=1 to 3;
    Do j=1 to 3-i;
      If c(j) < c(j+1) then do;
        Temp=c(j); c(j)=c(j+1);
        C(J+1)=temp;
      end;
    end;
  end;
Run;
```

This snippet of code results in a sorted list (descending) of blood pressure values.

QUICK SORT

The quick sort is also an algorithm designed to sort horizontally, first conceived by Tony Hoare in 1960. It is an early example of parallel processing in that it works by dividing a list of data values into successively smaller lists, iteratively choosing a pivot value, and continuously changing the array. It is more complex, and more efficient, than the bubble sort. QuickSort in SAS is a data step construct that mimics the Hoare's QuickSort algorithm. The code for SAS implementations is complex and verbose, therefore it is not demonstrated here. Please see an excellent paper by Paul Dorfman and a Wikipedia article on QuickSort, mentioned in the references section of the paper, for additional information.

The quicksort algorithm is extremely efficient for sorting arrays within larger data tables. The progressive division of the arrays into smaller and smaller segments makes the processing much faster than the bubble sort. However, coding a custom variation of the algorithm is arduous. Luckily, there are macros such as Paul Dorfman's Qsort macro that are readily available to modify. There are variants of SAS Quicksort to accommodate parallel arrays and other complex scenarios.

THREADED SORT

SAS has provided the option to take advantage of multiple processors in selected procedures since Version 9.2, improving performance. The THREADS system or procedure option allows parallel processing. The number of CPUs to use can also be specified. This enhances PROC SORT performance on large tables, increasing speed as the data set size increases.

The THREADS and NOTHEADS options turn on or off the ability to access multiple CPUs in selected SAS procedures including PROC SORT. These options can be applied on the system or procedure level. In addition, the CPUCOUNT system option allows you to select the number of CPUS.

Usage:

```
Options threads cpucount=4;
Proc sort data=yourdata; ...
```

To determine the number of CPUs on your system:

```
Proc options group=performance;
Run;
```

Check your log for the CPUCOUNT data.

SERPENTINE SORT

PROC SURVEYSELECT facilitates the drawing of a valid, stratified, random, proportional sample. PROC SURVEYSELECT utilizes either a nested sort (more commonly used in SAS) or a serpentine sort to stratify by strata (if requested) and then within strata sorts by control variables, swapping between ascending and descending order. A serpentine sort is the default (SORT=NEST, more commonly used in other SAS procedures, may also be requested). PROC SURVEYSELECT's serpentine sort arranges records by strata (if requested) and then within strata sorts by control variables, swapping between ascending and descending order. This method attempts to ensure robust representation of all values in all control variables in a sample.

Usage:

```
proc surveyselect data=tobesampled
method=sys seed=87654321
sampsize=(360 360 360 120) out=lib.sample01;
  strata sampcat;
  control census_region rural;
  id provider;
run;
```

Check your log for the Control Sorting information:

```
Control Sorting  Serpentine
```

YOU'VE GOT OPTIONS

Perhaps, like Harry and Neville, you are wondering if the SAS Sorting Hat was correct. We've looked at 4 magical methods to sort SAS data – now consider additional sort options and explore the beyond – the possibilities inherent in implicit sorting in SAS. Adding some sort options and/or preprocessing your data may reassure you that the hat chose wisely.

PROC SORT options may provide the answer to your remaining questions! Please see SAS documentation for more information on any of these options.

- Dealing with duplicates? Try NODUPKEY, NODUPRECS, DUPOUT=; NOUNIQUEKEY, UNIQUEOUT=

- Need to use a different ordering scheme? Try the DESCENDING, REVERSE, SORTSEQ=, standard collating sequences, ALTERNATE_HANDLING, STRENGTH options
- Dealing with large files? Use the TAGSORT, **THREADS**, **NOTHEADS**; SORTSIZE=
- Have multiple records per key? Try the EQUALS and NOEQUALS options
- Is your data already sorted? How do you know? Use the PRESORTED and FORCE options
- Does another method for the SORT-SORT-MERGE quandary make sense? Consider PROC SQL, PROC FORMAT, HASH, and INDICES
- Need to retain an original file date-time stamp on your output file? Try the DATECOPY option

IMPLICIT SORTING †††

A number of processes, SAS procedures, and SAS statements can perform implicit sorting and reorder records and variables in output data sets. It may not be necessary to use a sort method at all. Please see the SAS documentation for details on the implicit sorts listed below:

- CLASS STATEMENTS (numerous procedures and options)
- ORDER STATEMENTS (data, formatted, freq and internal)
- PROC SQL ORDER BY, GROUP BY, _METHOD
- BY NOTSORTED
- BY GROUPFORMAT
- Indices (Simple, Multiple, and Unique)
- Hash objects
- PRELOADFMT
- PROC TABULATE
- PROC REPORT ORDER, ORDER=

CONCLUSION

Lesser known and less frequently used SAS sorting techniques such as the bubble sort, the quick sort, the threaded sort and the serpentine sort can greatly enhance your SAS toolbox. Additionally, PROC SORT options and implicit sorts in SAS procedures may facilitate your work. Put on the SAS Sorting hat today and try these techniques out today!

REFERENCES

- Axelrod, Elizabeth. 2004. "Don't Resort to Re-sort". Proceedings of the Midwest SAS Users Group, Chicago, IL. https://www.lexjansen.com/mwsug/2004/Coders_Corner/C5_Axelrod.pdf
- Bachler, Robert. 1996. "PROC SQL as an Alternative to Multiple SORT/MERGE Statements". Proceedings of the Western Users of SAS Software, San Francisco, CA. <https://www.lexjansen.com/wuss/1996/WUSS96019.pdf>
- Bassett, Britta Kelsey. 2007. "The SORT Procedure: Beyond the Basics". Proceedings of the SAS Users Group International, San Francisco, CA. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi31/030-31.pdf>
- Bassett, Britta Kelsey. 2010. "The Mystery of the PROC SORT Options NODUPRECS and NODUPKEY Revealed". Proceedings of the Midwest SAS Users Group, Milwaukee, WI. <https://www.lexjansen.com/mwsug/2010/tutorials/MWSUG-2010-48.pdf>

Bavirisetty, Lakshmi Nirmala; Chaudhary, Kaushal; and Schreiber-Gregory, Deanna. 2017. "Hashtag #Efficiency! An Exploration of Hash Tables and Other Techniques". Proceedings of the PharmaSUG, Baltimore, MD. <https://www.lexjansen.com/pharmasug/2017/TT/PharmaSUG-2017-TT09.pdf>

Benjamin Jr., William E.. 2001. "Merging Disk Datasets to Large Unsorted Tape Datasets without a Sort Step, Using Indexed SAS Files". Proceedings of the SAS Users Group International, Long Beach, CA. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi26/p101-26.pdf>

Benjamin Jr., William E.. 2001. "Merging Small Datasets to Large Unsorted Tape Datasets without a Sort Step, Using SAS Formats". Proceedings of the SAS Users Group International, Long Beach, CA. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi26/p102-26.pdf>

Buck, Janice M.B.. 1987. "Comparison of PROC SORTT, Syncsort, and Plsort". Proceedings of the SAS Users Group International, Dallas, TX. <https://support.sas.com/resources/papers/proceedings-archive/SUGI87/Sugi-12-176 Buck.pdf>

Cassell, David L.. 2005. "A Sort of a Mess ?? Sorting Large Datasets on Multiple Keys". Proceedings of the Pacific Northwest SAS Users Group, Tacoma, WA. <https://www.lexjansen.com/pnwsg/2005/David Cassell - A Sort Of Mess.pdf>

Chao, Szu-Fu. 2017. "Identifying and Removing Subjects with Duplicates in Long Data Format: A Simple Approach Using PROC SORT and PROC SQL". Proceedings of the South East SAS Users Group, Cary, NC. http://analytics.ncsu.edu/sesug/2017/SESUG2017_Paper-116_Final_PDF.pdf

Cherny, Max. 2015. "Getting the Most Out of PROC SORT: A Review of Its Advanced Options". Proceedings of the PharmaSUG, Orlando, FL. <https://www.lexjansen.com/pharmasug/2015/QT/PharmaSUG-2015-QT14.pdf>

Cisternas, Miriam. 2001. "How Using Arrays and Formats Allowed Me to Draw 16,000 Random Samples of Specific N without Sorting (and save my hard disk and sanity, too)". Proceedings of the Western Users of SAS Software, San Francisco, CA. <https://www.lexjansen.com/wuss/2001/WUSS01046.pdf>

Dorfman, Paul M.. 1999. "Alternative Approach to Sorting Arrays and Strings: Tuned DATA Step Implementations of Quicksort and Distribution Counting". Proceedings of the SAS Users Group International, Miami Beach, FL. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi24/Posters/p232-24.pdf>

Dorfman, Paul M.. 2001. "Quick Sorting an Array". Proceedings of the SAS Users Group International, Long Beach, CA. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi26/p096-26.pdf>

Dorfman, Paul M.; and Shajenko, Lessia S.. 2006. "Already Sorted? Make Good Use of It!". Proceedings of the South East SAS Users Group, Atlanta, GA. http://analytics.ncsu.edu/sesug/2006/SC10_06.PDF

Dorfman, Paul. 2018. "Sorting Arrays Using the Hash Object". Proceedings of the South East SAS Users Group, St. Petersburg, FL. https://www.lexjansen.com/sesug/2018/SESUG2018_Paper-288_Final_PDF.pdf

Fickbohm, David. 2007. "The SORT Procedure: Beyond the Basics". Proceedings of the Western Users of SAS Software, San Francisco, CA. https://www.lexjansen.com/wuss/2007/ApplicationsDevelopment/APP_Fickbaum_SortPrcedure.pdf

Fine, Lisa. 2011. "ORDER, ORDER PLEASE: SORTING DATA USING PROC REPORT". Proceedings of the SAS Global Forum, Las Vegas, NV. <https://support.sas.com/resources/papers/proceedings11/090-2011.pdf>

First, Steven. 2010. "A Different View of PROC SORT". Proceedings of the Midwest SAS Users Group, Milwaukee, WI. <https://www.lexjansen.com/mwsug/2010/advanced/MW/SUG-2010-158.pdf>

Frye, Barry. 2017. "Data Validation Using the SAS® SORT Procedure and MERGE Statement". Proceedings of the SAS Global Forum, Orlando, FL. <https://support.sas.com/resources/papers/proceedings17/0886-2017.pdf>

Groeneveld, Jim. 2006. "Sorting by formatted values". Proceedings of the Pharmaceutical Users Software Exchange, Dublin, Ireland. <https://www.lexjansen.com/phuse/2006/cc/CC05.pdf>

Hadden, Louise. 2005. "PROC SURVEYSELECT: A Simply Serpentine Solution for Complex Sample Designs". Proceedings of the Northeast SAS Users Group, Portland, ME.
<https://www.lexjansen.com/nesug/nesug05/an/an5.pdf>

Hamburger, Merle E.; and Sukalac, Thomas. 2005. "Linking Unlinked Data: PROCs SQL and SORT Can Do It". Proceedings of the SAS Users Group International, Montreal, PQ.
<https://support.sas.com/resources/papers/proceedings/proceedings/sugi29/079-29.pdf>

Hopkins, David. 1994. "Another Sorting Technique for Large, Troublesome Datasets". Proceedings of the South East SAS Users Group, Charleston, SC. <https://www.lexjansen.com/sesug/1994/SESUG94020.pdf>

Hughes, Troy Martin. 2016. "Sorting a Billion Records: Conquering Scalability in a Big Data World". Proceedings of the SAS Global Forum, Las Vegas, NV.
<https://support.sas.com/resources/papers/proceedings16/11888-2016.pdf>

Jablonski, Bartosz. 2019. "Use advantage of INDEXes even if WHERE clause contains OR condition". Proceedings of the SAS Global Forum, Dallas, TX.
<https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3722-2019.pdf>

Jain, Adish; and Bachtell, Kate. 2012. "Reordering Values within Observations: Beyond CALL SORTC(N)". Proceedings of the SAS Global Forum, Orlando, FL.
<https://support.sas.com/resources/papers/proceedings12/117-2012.pdf>

Jolley, Linda; and Stroupe, Jane. 2010. "Dear Miss SASAnswers: A Guide to Sorting Your Data". Proceedings of the SAS Global Forum, Seattle, WA.
<https://support.sas.com/resources/papers/proceedings10/140-2010.pdf>

Kale, Sanket. 2010. "Sort Your Problems- Using SORTEDBY for Data Compares". Proceedings of the PharmaSUG, Orlando, FL. <https://www.lexjansen.com/pharmasug/2010/DM/DM07.pdf>

Keintz, Mark. 2009. "A Faster Index for Sorted SAS® Datasets". Proceedings of the SAS Global Forum, National Harbor, MD. <https://support.sas.com/resources/papers/proceedings09/024-2009.pdf>

Keintz, Mark. 2012. "Condensed and Sparse Indexes for Sorted SAS® Datasets". Proceedings of the Northeast SAS Users Group, Baltimore, MD. <https://www.lexjansen.com/nesug/nesug12/ld/ld06.pdf>

Keintz, Mark; and Zhang, Shuguang. 2007. "Outperforming SAS® Indices for Sorted Datasets". Proceedings of the Northeast SAS Users Group, Baltimore, MD.
<https://www.lexjansen.com/nesug/nesug07/bb/bb06.pdf>

Khudov, Vladyslav. 2017. "Are you Still Afraid of Using Arrays? Let's Explore their Advantages". Proceedings of the Pharmaceutical Users Software Exchange, Edinburgh, Scotland.
<https://www.lexjansen.com/phuse/2017/ct/CT07.pdf>

Kuligowski, Andrew T.. 2015. "Looking Beneath the Surface of Sorting". Proceedings of the PharmaSUG, Orlando, FL. <https://www.lexjansen.com/pharmasug/2015/TT/PharmaSUG-2015-TT13.pdf>

Kuligowski, Andrew. 2010. "The Ins and Outs and Ups and Downs of Sorted Data". Proceedings of the PharmaSUG, Orlando, FL. <https://www.lexjansen.com/pharmasug/2010/TU/TU07.pdf>

LaBrecque, Leslie. 1997. "A Method for Sorting a Large Data Set with Limited Memory". Proceedings of the Northeast SAS Users Group, Baltimore, MD.
<https://www.lexjansen.com/nesug/nesug97/coders/labrecqu.pdf>

Lavery, Russ. 2013. "Fast Access Tricks for Large Sorted SAS Files". Proceedings of the Midwest SAS Users Group, Columbus, OH. <https://www.lexjansen.com/mwsug/2013/HW/MWSUG-2013-HW02.pdf>

Lin, Amanda; and Jia, Justin. 2013. "Horizontal Data Sorting and Insightful Reporting: A Useful SAS® Technique". Proceedings of the SAS Global Forum, San Francisco, CA.
<https://support.sas.com/resources/papers/proceedings13/376-2013.pdf>

Lund, Peter. 1998. "A Simple Solution to the PROC SORT NOPUD Option Problem". Proceedings of the Pacific Northwest SAS Users Group, Portland, OR.
<https://www.lexjansen.com/pnwsug/1998/PNWSUG98025.pdf>

Mahan, Robert L.. 1988. "Sort Efficiency Using the SAS® System Under VM/CMS". Proceedings of the SAS Users Group International, Orlando, FL. https://support.sas.com/resources/papers/proceedings-archive/SUGI88/Sugi-13-22_Mahan.pdf

Markovitz, Heidi. 2006. "Dup, Dedup, DUPOUT - New in PROC SORT". Proceedings of the South East SAS Users Group, Atlanta, GA. http://analytics.ncsu.edu/sesug/2006/CC14_06.PDF

Mebust, Scott. 2006. "Getting the Best Performance from SAS®9 Threaded PROC SORT". Proceedings of the SAS Users Group International, Philadelphia, PA. <http://support.sas.com/rnd/papers/sugi30/V9SORT.ppt>

Miller, Kevin. 2018. "Get Your Sort On: Macro Driven Method for Automated Unique Sorting". Proceedings of the PharmaSUG, Seattle, WA. <https://www.lexjansen.com/pharmasug/2018/BB/PharmaSUG-2018-BB15.pdf>

Milum, Jenine. 2012. "PROC FORMAT, a Speedy Alternative to Sort/Sort/Merge". Proceedings of the SAS Global Forum, Orlando, FL. <https://support.sas.com/resources/papers/proceedings12/428-2012.pdf>

Morgan, Derek. 2018. "PROC SORT (then and) NOW". Proceedings of the SAS Global Forum, Denver, CO. <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2773-2018.pdf>

Mullins, Lynn; and Watson, Richann. 2017. "Exploring HASH Tables vs. SORT/DATA Step vs. PROC SQL". Proceedings of the Midwest SAS Users Group, St. Louis, MO. <https://www.lexjansen.com/mwsug/2017/BB/MWSUG-2017-BB124.pdf>

Neft, Matthew; and Pronko, Chelle. 2014. "Using Sorting Algorithms to Create Sorted Lists". Proceedings of the SAS Global Forum, Washington, DC. <https://support.sas.com/resources/papers/proceedings14/2025-2014.pdf>

Nunez, Kirsty. 1989. "Comparison of VMSORT, SyncSort, PLSORT and SORTT under VM/CMS Using Release 5.18 of the SAS® System". Proceedings of the SAS Users Group International, San Francisco, CA. https://support.sas.com/resources/papers/proceedings-archive/SUGI89/Sugi-89-86_Nunez.pdf

Price, Karen. 2014. "Move over MERGE, SQL and SORT. There is a faster game in town! #Hash Table". Proceedings of the South East SAS Users Group, Myrtle Beach, SC. <http://analytics.ncsu.edu/sesug/2014/BB-11.pdf>

Raithel, Michael A.. 1999. "What Sort of Input Should You Input to a Sort?". Proceedings of the SAS Users Group International, Miami Beach, FL. <https://support.sas.com/resources/papers/proceedings/proceedings/sugi24/Coders/p095-24.pdf>

Retzlaff, Tommy L.. 2007. "Sorting Out the Possibilities Using Retain". Proceedings of the PharmaSUG, Denver, CO. <https://www.lexjansen.com/pharmasug/2007/cc/CC25.pdf>

Rubin, David S.. 1988. "Sorting Massive Data Sets". Proceedings of the SAS Users Group International, Orlando, FL. https://support.sas.com/resources/papers/proceedings-archive/SUGI88/Sugi-13-244_Rubin.pdf

Samudral, Karuna; and Giddings, Gregory. 2006. "By's Notsorted Option". Proceedings of the Northeast SAS Users Group, Philadelphia, PA. <https://www.lexjansen.com/nesug/nesug06/po/po14.pdf>

Sevick, Carter. 2007. "Automating the SORT-SORT-MERGE and Overcoming the Many to Many MERGE: %ProcMerge". Proceedings of the Western Users of SAS Software, San Francisco, CA. https://www.lexjansen.com/wuss/2007/Tutorials/TUT_Sevick_AutomatingSORTSORTMERGE.pdf

Shliakhov, Daniil. 2015. "Sorting big datasets. Do we really need it?". Proceedings of the PharmaSUG, Orlando, FL. <https://www.lexjansen.com/pharmasug/2015/QT/PharmaSUG-2015-QT21.pdf>

Shu, Haibin; and Rojco, Elena. 2015. "Fifty Shades of Sorting". Proceedings of the PharmaSUG China, Orlando, FL. <https://www.lexjansen.com/pharmasug-cn/2015/PT/PharmaSUG-China-2015-PT02.pdf>

Smith, Jason A.. 2016. "Perish the Sort: Using Indexes and Hash Objects for Efficient Programming". Proceedings of the Pharmaceutical Users Software Exchange, Barcelona, Spain.
<https://www.lexjansen.com/phuse/2016/cc/CC08.pdf>

Sridharma, Selvaratnam. 2012. "Sorting a Large Data Set When Space is Limited". Proceedings of the Northeast SAS Users Group, Baltimore, MD. <https://www.lexjansen.com/nesug/nesug12/cc/cc36.pdf>

Virgile, Bob. 2001. "So You're Running Out of Sort Work Space . . . Reducing the Resources Used by PROC SORT". Proceedings of the Northeast SAS Users Group, Portland, ME.
<https://www.lexjansen.com/nesug/nesug01/at/at1015.pdf>

Volya, Rita. 2013. "Alternative Methods for Sorting Large Files without Leaving a Big Disk Space Footprint". Proceedings of the Northeast SAS Users Group, Burlington, VT.
https://www.lexjansen.com/nesug/nesug13/29_Final_Paper.pdf

Wainwright-Zimmerman,. 2012. "Intelligent PROC SORT NODUPKEY". Proceedings of the SAS Global Forum, Orlando, FL. <https://support.sas.com/resources/papers/proceedings12/041-2012.pdf>

Wang, Tim QF. 2010. "Merging Historical Data Automatically While Dynamically Sorting the Variables". Proceedings of the South Central SAS Users Group, Austin, TX.
https://www.lexjansen.com/scsug/2010/Wang/TimWang_Sort_variables.pdf

Yan, Lin. 2004. "A Quicker Way to Inputting and Sorting Large Dataset Using SAS". Proceedings of the PharmaSUG, San Diego, CA.
<https://www.lexjansen.com/pharmasug/2004/ApplicationsDevelopment/AD16.pdf>

ACKNOWLEDGMENTS

The authors wish to thank Paul Dorfman, Troy Martin Hughes, Justin Jia, Amanda Lin, Matthew Neff, Chelle Pronko, Ron Fehd (Macro Maven), and Henri Theuwissen for their illuminating manuscripts on sorting techniques in SAS.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Charu Shankar
SAS Institute Canada
Charu.Shankar@sas.com

Louise S. Hadden
Abt Associates Inc.
Louise_hadden@abtassoc.com