

ACM Algorithms Policy

FRED T. KROGH



MATTERS OF CONTENT

A contribution should be in the form of an algorithm, comparison, certification, remark, or translation.

Algorithms

ACM Algorithms are published to make the fruits of software research readily available to as wide an audience as possible. An algorithm must either provide a capability not readily available or perform a task better in some way than has been done before. “Better” can mean anything from improved reliability or efficiency to more attractive packaging. In all cases, an algorithm must be of lasting value and must represent a substantial contribution in terms of the amount of work or the originality required for its creation.

In most cases the communication of new algorithmic ideas should be done either in a companion paper published in one of the ACM Transactions or in previously published work. The textual part of an algorithm submission should give a brief description of what the algorithm does and pertinent information on usage and maintenance. It should not duplicate information in another paper or in the algorithm listing. It is recognized that certain algorithms will be useful to readers of one of the ACM Transactions but deal with subjects not in the usual research domain of that journal. When appropriate, the Algorithms Editor will guide authors to other ACM journals for possible publication of such algorithms.

An ACM algorithm must be complete, portable, well documented, and well structured. The meaning of these terms is clarified below.

1. *Completeness.* A submission must include all of the code and test data necessary for the effective use and testing of the algorithm by a large segment of its intended audience. To assist those who use the algorithm, the author should provide a small test driver that illustrates the use of the algorithm for a simple test problem. For testing purposes, one should provide in a single driver a sufficient variety of test cases to exercise all the main features of the code. All submitted code, including test drivers and preprocessors, is subject to the refereeing process. Code subject to more restricted use than specified in this policy may be used in a supporting role for an algorithm provided it is available from an established source for a nominal fee, and there is no nearly equivalent code available in the public domain.

2. *Portability.* It must be possible to move the code in machine-readable form from one machine to another with only minor, well-documented changes. Programs written in a language having several popular dialects should use only language features common to those dialects. As initial evidence of portability, the author either can include evidence of successful execution on three different

computers, that is, computers with different basic instruction sets, or better, when available, can give the result of running the software through a verifier that checks the code for portability or conformance to a standard. All FORTRAN programs must conform to the ANSI X3.9-1978 FORTRAN 77 [1].

Machine-dependent modules, including assembly language, may be used provided:

- (a) They are clearly specified, limited in function, and either necessary or a substantial improvement over equivalent portable procedures.
- (b) Where possible, portable versions are also submitted to facilitate installation and testing, and to confirm the reader's understanding of the specifications. If portable versions are not provided, then tested versions for at least three different computers must be provided.
- (c) A portable test program, which exhaustively tests each machine-dependent module of the package, is provided.

3. *Documentation.* Each module of the code, including test drivers, must be adequately commented. Comments should include the purpose of the module, definitions of all arguments passed through the calling sequence, through global variable declarations, or obtained via input, and comments setting off and explaining major parts of the code. Comments defining machine dependencies should be gathered together and clearly flagged. Standard names and definitions for machine constants are normally to be used [2]. If these constants are used widely in the submitted algorithms, then they may be accessed through a procedure [3]. Other constants clearly identified in comments (e.g., BIG = SQRT(SOVFLO)) and machine dependencies not in the standard [3] may be introduced. A comment can simply point to the place where full comments are given in cases where large comment blocks would otherwise be repeated. An alphabetical list of internal variables, together with how they are used, is highly recommended. (Identify temporary variables as temporary.) Machine-readable documentation giving more detailed information about the package than can be published is encouraged.

4. *Structure.* Code should be organized so that it is easy to use and modify, and yet is flexible enough to be useful for most problems in the problem area covered. Indentation should be used to identify loops, compound statements and blocks, and continuation of a statement onto another line. If integers are used for statement labels, they must be strictly increasing and should be far enough apart to allow for future modifications. If the space required by an array is highly problem dependent, then that array should have a variable dimension. (NOTE:FORTRAN requires common blocks with the same name to have the same length.)

Algorithmic ideas may be published as integral parts of regular papers without meeting these criteria. This provision may not, however, be used as a subterfuge to circumvent meeting the criteria for ACM algorithms.

All contributions will appear in their entirety in ACM's looseleaf algorithms periodical *Collected Algorithms from ACM* (CALGO). Algorithms are available in machine-readable form through the ACM Algorithms Distribution Service.

Comparisons

A comparison is a report on the relative merits and features of highly similar software packages for a specific subject area. This study normally includes reporting and interpreting various cogent observed facts about the packages. These facts are typically based on solving a common set of test problems. The drivers, test problems, and data used in the study are to be sent to the Algorithms Editor, at the time of submission, as an algorithm. This body of information will be used in the refereeing process and will be available as an ACM Algorithm.

Certifications

A certification is a report on a previously published algorithm. It can be a careful study of performance characteristics, a verification of correctness, or a report on extensive testing.

Remarks

A remark is a brief report on a previously published algorithm. It is usually concerned with corrections or modifications.

Translations

A translation may provide either machine-dependent modules for a machine or operating system not covered by an ACM algorithm, or a translation of the algorithm into a different high-level language. A translation will only be considered if it is a translation of an algorithm that still represents the current state of the art and if it satisfies all the criteria of a regular algorithm submission.

SUBMISSION FORMALITIES

1. Five copies of all textual material should be sent to the Algorithms Editor. This material must be typewritten and double spaced. Companion papers to algorithms submitted to one of the ACM Transactions should be sent with the algorithm to the Algorithms Editor, as they will be refereed along with the algorithm. The purpose of those parts of the code submission that are included for testing purposes or serve only a supporting role should be clearly identified.
2. Processing of an algorithm will not proceed until it is verified that machine-readable copy of the algorithm has arrived in good form. Before sending machine-readable copy, contact the Algorithms Editor and explain the computer(s) available to you, the size of your submission, and in what form you would like to send the material. If an algorithm requires the use of code subject to restricted use, the use of this code should be cleared first with the Algorithms Editor.
3. Evidence of portability must be included with an algorithm submission.

Dissemination Agreement

Submittal of an algorithm for publication in one of the ACM Transactions implies that unrestricted use of the algorithm within a computer is permissible. General permission to copy and distribute the algorithm without fee is granted provided that the copies are not made or distributed for direct commercial advantage. The ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association

for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Copyright Agreement

Authors of copyrightable algorithms (or their employers) are required to transfer the copyright to ACM upon acceptance of the algorithm for publication, in accordance with ACM policy to own copyright on ACM published material. The Association grants to authors the right to reuse their material and seeks the permission of authors before approving republication.

REFERENCES

1. AMERICAN NATIONAL STANDARDS INSTITUTE. American National Standard programming language FORTRAN, ANSI X3.9-1978, American National Standards Institute, New York, 1978.
2. FORD, B. Parameterization of the environment for transportable numerical software. *ACM Trans. Math. Softw.* 4, 2 (June 1978), 100–103.
3. FOX, P. A., HALL, A. D., AND SCHRYER, N. L. Algorithm 528: Framework for a portable library. *ACM Trans. Math. Softw.* 4, 2 (June 1978), 177–188.