Bubble Sort
An Archaelogical Algorithmic Analysis

Owen Astrachan

ola@cs.duke.edu

http://www.cs.duke.edu/~ola

NSF CAREER 9702550 CRCD 0088078

Bubble sort: what can we study?

- What is bubble sort and what are its origins?
  - When did it appear, when was it named?
- Why is bubble sort studied and why is it popular?
  - Do bubble sorters fly in the face of expert opinion?
- What do the experts say about bubble sort?
  - What makes someone an expert?
    - Turing, Grace Hopper, Karlstrom, SIGCSE Award winners
- What do we learn from algorithm archaelogy?
  - Libraries, Google, and the Internet are amazing

Why Bubble is not my Favorite Sort

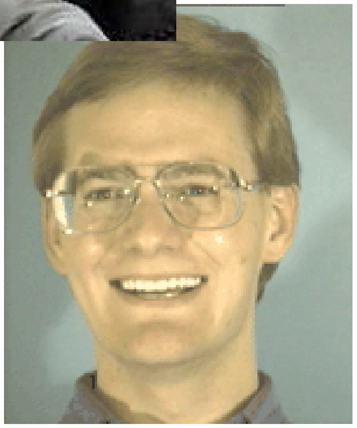
- 1973: Wrote my first program
  - Biorhythms in BASIC
- 1975: First formal CS course
  - Programming as trial and error, no elegance
- 1977: Sysprog 'ls' in DXPL
  - I use bubble sort!
- 1980: Teach using Apple II
  - High school BASIC
- 1985: Teach using IBM PC
  - College Pascal

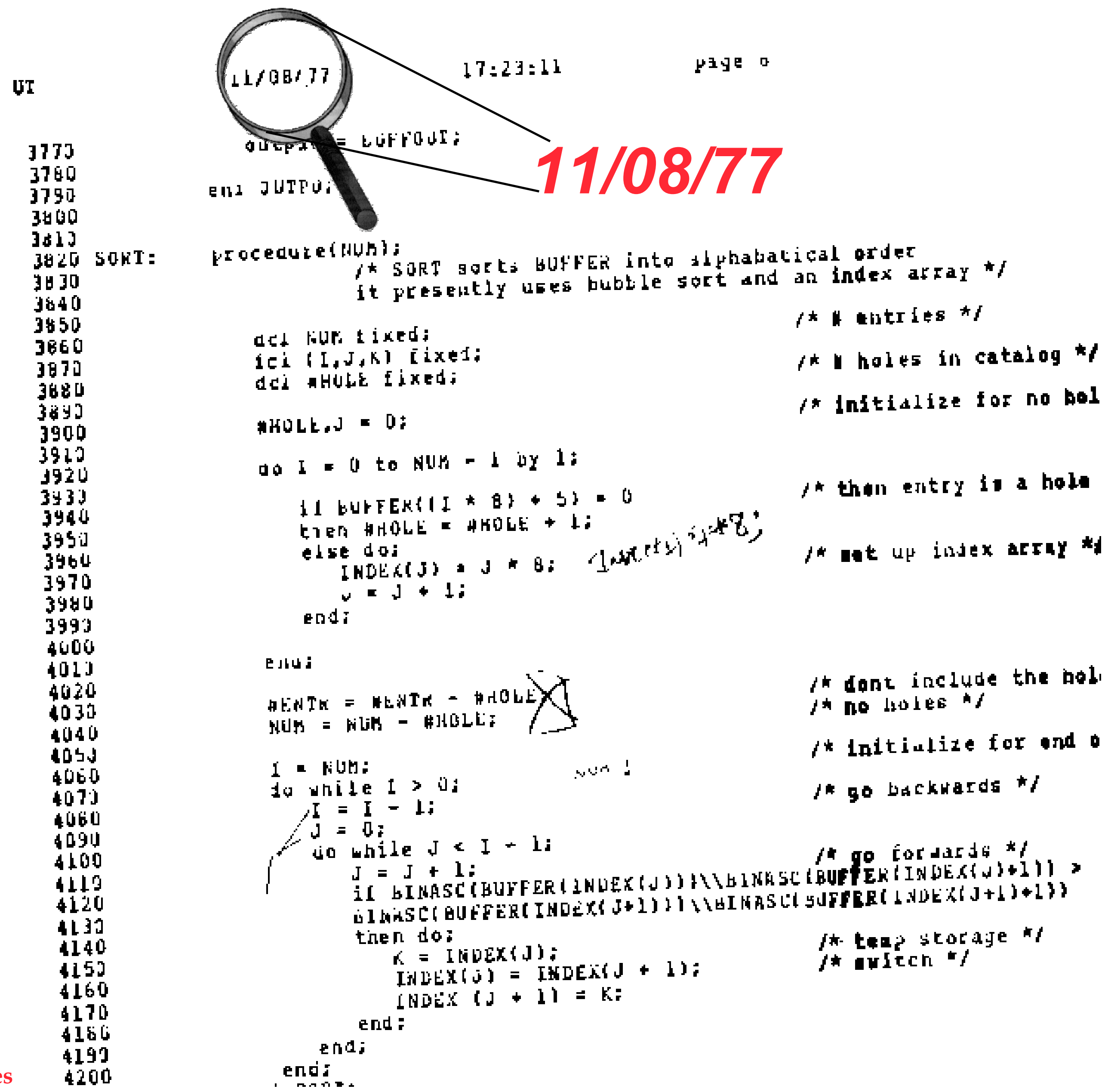
Genesis of Paper

Wolfman & Pottinger (Ernst)

Taste's great!, Less filling!

[I recall] 'A human sorting algorithm in which students were to sort themselves by ID using bubble sort --- the instructor set us off with an exuberant "Go!" After several frantic seconds of inter-student hubbub and instructor protestation we were sorted, leaving the Great Master lamenting that "only one person gets to talk at a time!!" '

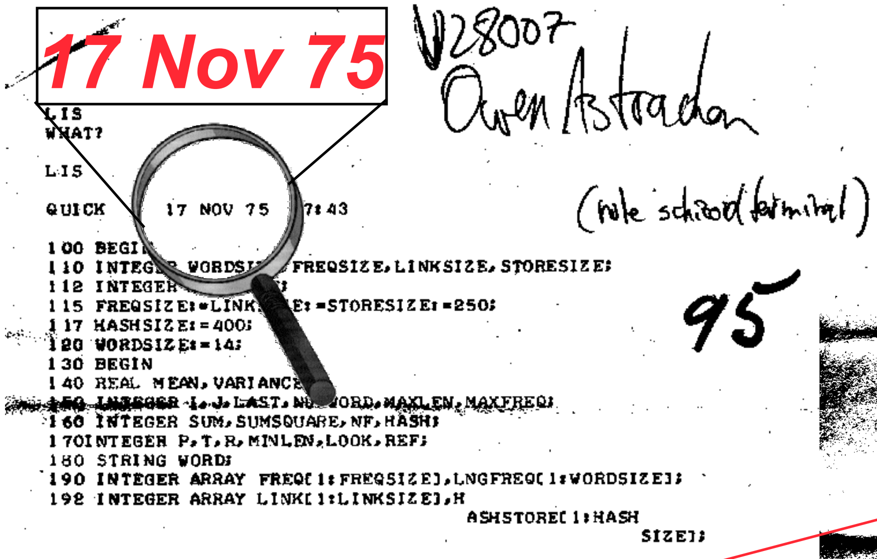




11/08/77

```
UT                11/08/77              17:23:11              page 6

                    output = BUFFOUT;

                 end OUTPUT;

       SORT:     procedure(NUM);
                              /* SORT sorts BUFFER into alphabatical order
                              it presently uses bubble sort and an index array */
                                                                   /* # entries */
                     dcl NUM fixed;
                     dcl (I,J,K) fixed;                            /* # holes in catalog */
                     dcl #HOLE fixed;
                                                                   /* initialize for no hol
                     #HOLE,J = 0;

                     do I = 0 to NUM - 1 by 1;
                                                                   /* then entry is a hole
                        if BUFFER((I * 8) + 5) = 0
                        then #HOLE = #HOLE + 1;     INDEX(J) = J*8;
                        else do;                                   /* set up index array */
                           INDEX(J) = J * 8;
                           J = J + 1;
                        end;

                     end;
                                                                   /* dont include the hol
                     #ENTR = #ENTR - #HOLE;                        /* no holes */
                     NUM = NUM - #HOLE;
                                                                   /* initialize for end o
                     I = NUM;                    NUM 1
                     do while I > 0;                               /* go backwards */
                        I = I - 1;
                        J = 0;
                        do while J < I - 1;                        /* go forwards */
                           J = J + 1;
                           if BINASC(BUFFER(INDEX(J)))\\BINASC(BUFFER(INDEX(J)+1)) >
                           BINASC(BUFFER(INDEX(J+1)))\\BINASC(BUFFER(INDEX(J+1)+1))
                           then do;                                /* temp storage */
                              K = INDEX(J);                        /* switch */
                              INDEX(J) = INDEX(J + 1);
                              INDEX (J + 1) = K;
                           end;
                        end;
                     end;
```

17 Nov 75

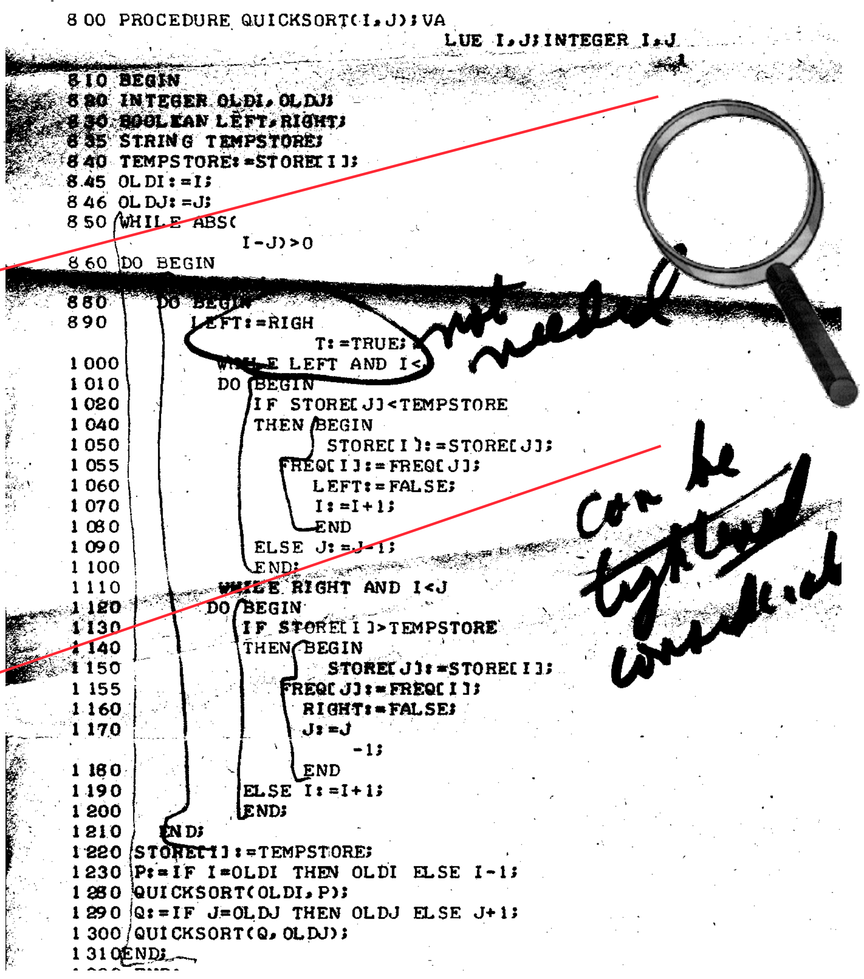


U28007
Owen Astrachan

(note schizoid terminal)

95

```
LIS
WHAT?

LIS

QUICK        17 NOV 75    7:43

100 BEGIN
110 INTEGER WORDSIZE,FREQSIZE,LINKSIZE,STORESIZE;
112 INTEGER      ;
115 FREQSIZE:=LINKSIZE:=STORESIZE:=250;
117 HASHSIZE:=400;
120 WORDSIZE:=14;
130 BEGIN
140 REAL MEAN,VARIANCE
150 INTEGER I,J,LAST,NEWWORD,MAXLEN,MAXFREQ;
160 INTEGER SUM,SUMSQUARE,NF,HASH;
170INTEGER P,T,R,MINLEN,LOOK,REF;
180 STRING WORD;
190 INTEGER ARRAY FREQ[1:FREQSIZE],LNGFREQ[1:WORDSIZE];
192 INTEGER ARRAY LINK[1:LINKSIZE],H
                                        ASHSTORE[1:HASH
                                                   SIZE];
```



```
800 PROCEDURE QUICKSORT(I,J);VA
                                  LUE I,J;INTEGER I,J
810 BEGIN
820 INTEGER OLDI,OLDJ;
830 BOOLEAN LEFT,RIGHT;
835 STRING TEMPSTORE;
840 TEMPSTORE:=STORE[I];
845 OLDI:=I;
846 OLDJ:=J;
850 WHILE ABS(
                  I-J)>0
860 DO BEGIN
880      DO BEGIN
890         LEFT:=RIGH
                      T:=TRUE;
1000        WHILE LEFT AND I<
1010          DO BEGIN
1020             IF STORE[J]<TEMPSTORE
1040             THEN BEGIN
1050                   STORE[I]:=STORE[J];
1055               FREQ[I]:=FREQ[J];
1060                  LEFT:=FALSE;
1070                  I:=I+1;
1080                  END
1090             ELSE J:=J-1;
1100             END;
1110         WHILE RIGHT AND I<J
1120         DO BEGIN
1130            IF STORE[I]>TEMPSTORE
1140            THEN BEGIN
1150                  STORE[J]:=STORE[I];
1155              FREQ[J]:=FREQ[I];
1160                RIGHT:=FALSE;
1170                J:=J
                        -1;
1180                END
1190            ELSE I:=I+1;
1200            END;
1210     END;
1220 STORE[I]:=TEMPSTORE;
1230 P:=IF I=OLDI THEN OLDI ELSE I-1;
1280 QUICKSORT(OLDI,P);
1290 Q:=IF J=OLDJ THEN OLDJ ELSE J+1;
1300 QUICKSORT(Q,OLDJ);
1310END;
```

not needed

can be tightened considerably

Not needed

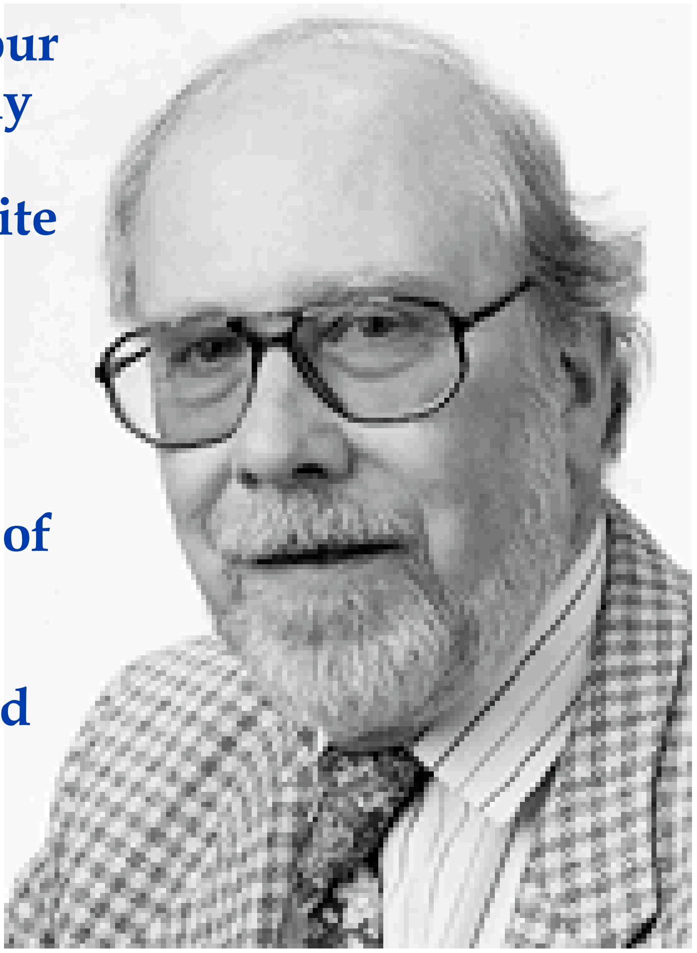
Can be tightened considerably

Bubblesort

Niklaus Wirth (Turing award 1984)

I have read your article and share your view that Bubble Sort has hardly any merits. I think that it is so often mentioned, because it illustrates quite well the principle of sorting by exchanging.

I think BS is popular, because it fits well into a systematic development of sorting algorithms. But it plays no role in actual applications. Quite in contrast to C, also without merit (and its derivative Java), among programming codes.

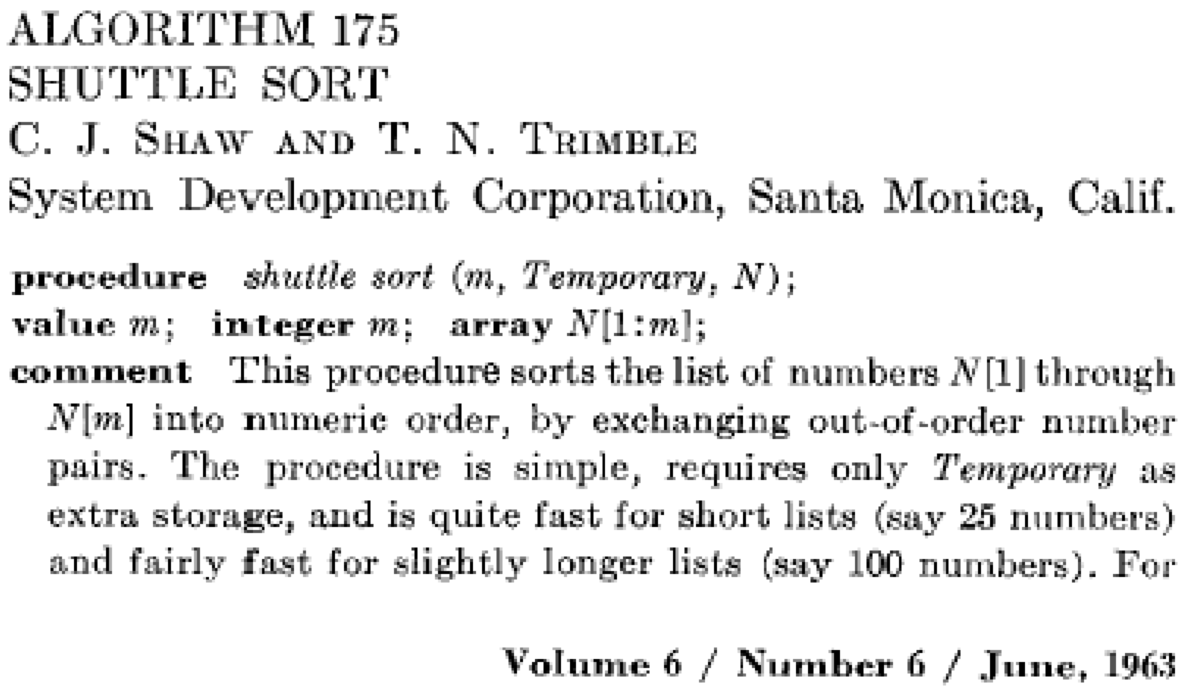


Bubble Sort: The code

```
void BubbleSort(Vector a, int n)
{
    for(int j=n-1; j > 0; j--)
        for(int k=0; k < j; k++)
            if (a[k+1] < a[k])
                Swap(a,k,k+1);
}
```

- What happens if `k+1` becomes `j` ?
- How do we stop early?
- What about alternating directions?
- Correctness: terminating, sorted, permutation.

Shuttle/Bubble becomes official

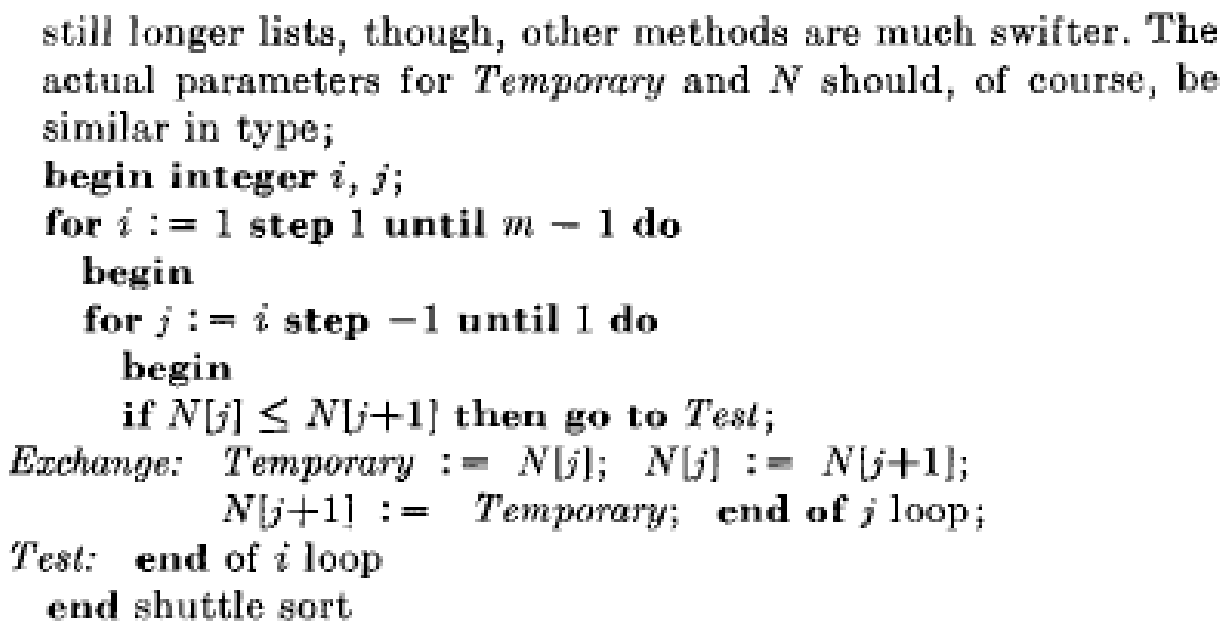


ALGORITHM 175
SHUTTLE SORT
C. J. Shaw and T. N. Trimble
System Development Corporation, Santa Monica, Calif.

procedure *shuttle sort* (m, *Temporary*, N);
value m; **integer** m; **array** $N[1:m]$;
comment This procedure sorts the list of numbers $N[1]$ through $N[m]$ into numeric order, by exchanging out-of-order number pairs. The procedure is simple, requires only *Temporary* as extra storage, and is quite fast for short lists (say 25 numbers) and fairly fast for slightly longer lists (say 100 numbers). For

Volume 6 / Number 6 / June, 1963

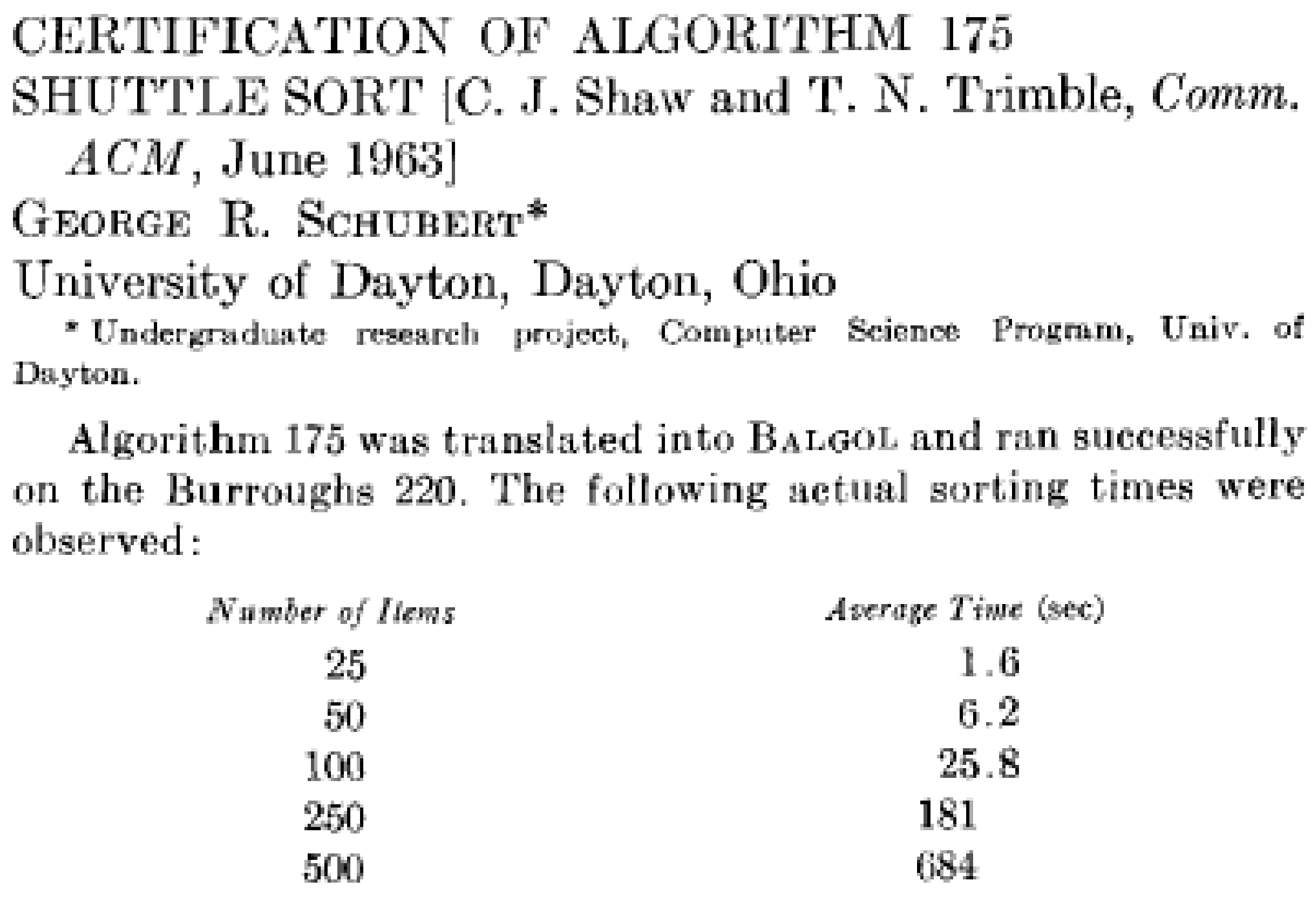
Algorithm 175: The code



still longer lists, though, other methods are much swifter. The actual parameters for *Temporary* and *N* should, of course, be similar in type;

```
begin integer i, j;
for i := 1 step 1 until m − 1 do
   begin
   for j := i step −1 until 1 do
      begin
      if N[j] ≤ N[j+1] then go to Test;
Exchange:  Temporary := N[j];  N[j] := N[j+1];
           N[j+1] := Temporary;  end of j loop;
Test:  end of i loop
  end shuttle sort
```

Certification of Algorithm 175



CERTIFICATION OF ALGORITHM 175
SHUTTLE SORT [C. J. Shaw and T. N. Trimble, *Comm. ACM*, June 1963]

GEORGE R. SCHUBERT*
University of Dayton, Dayton, Ohio

* Undergraduate research project, Computer Science Program, Univ. of Dayton.

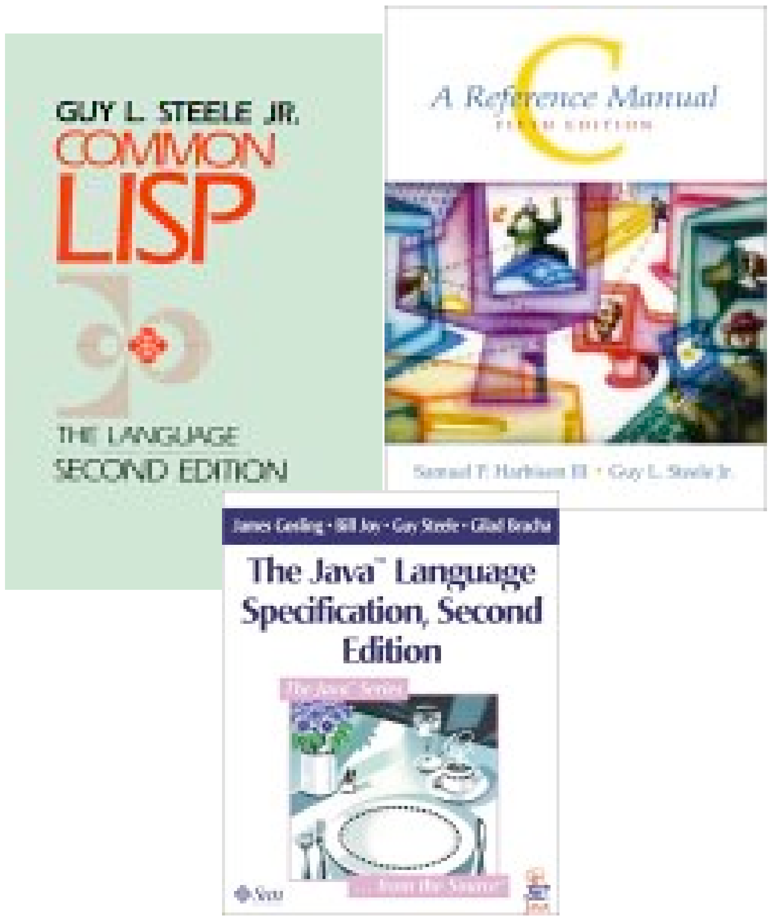
Algorithm 175 was translated into BALGOL and ran successfully on the Burroughs 220. The following actual sorting times were observed:

Number of Items	*Average Time* (sec)
25	1.6
50	6.2
100	25.8
250	181
500	684

Guy L. Steele, Jr. (Hopper '88)

(Thank you for your fascinating paper and inquiry. Here are some off-the-cuff thoughts on the subject.)

I think that one reason for the popularity of Bubble Sort is that it is easy to see why it works, and the idea is simple enough that one can carry it around in one's head …


GUY L. STEELE JR.
COMMON LISP
THE LANGUAGE
SECOND EDITION
C
A Reference Manual
FIFTH EDITION
Samuel P. Harbison III · Guy L. Steele Jr.
James Gosling · Bill Joy · Guy Steele · Gilad Bracha
The Java™ Language Specification, Second Edition
The Java™ Series
...from the Source™
Sun

continued

Guy L. Steele, Jr. (continued)

- When I was a teenager, I knew about and sometimes used the following "even worse" version of Bubble Sort:

```
do {
      swapflag = false;
      for (int j=n-1; j > 0; j--) {
            if (a[j+1] < a[j]) {
                  Swap(a,j,j+1);
                  swapflag = true;
            }
      }
   } while (swapflag);
```

- I no longer remember on what street corner I picked this up.

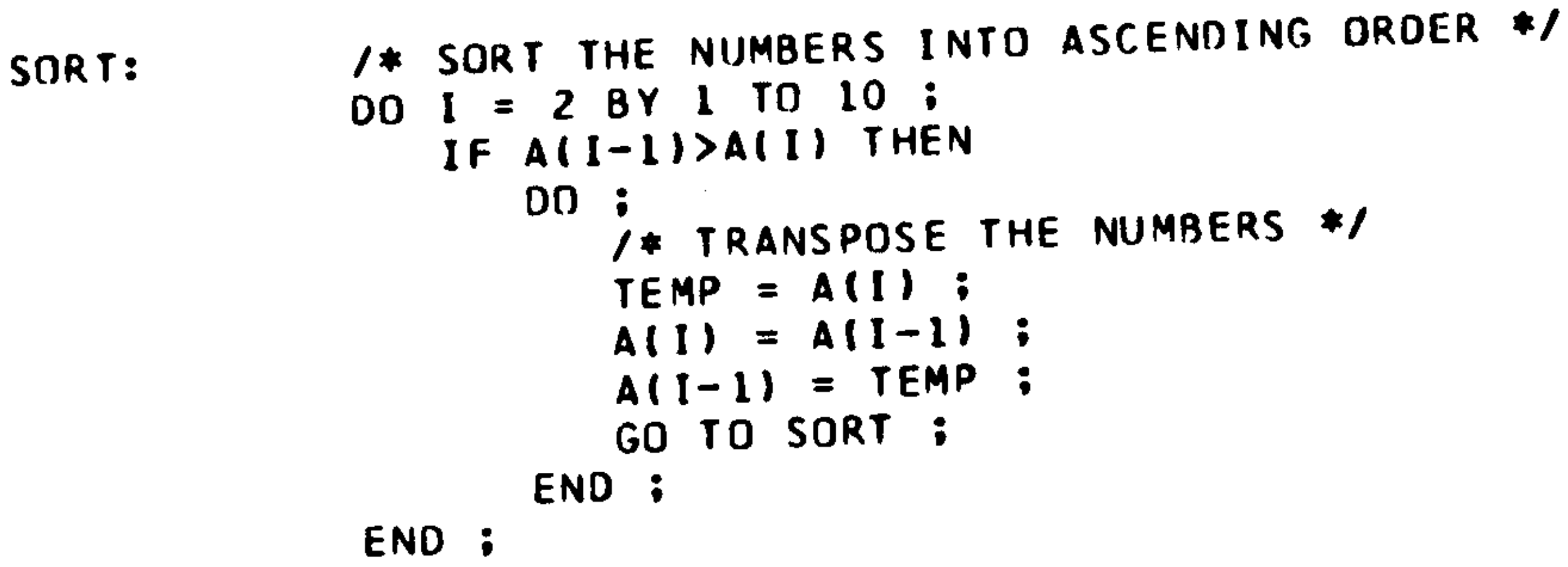
Guy L. Steele, Jr.

As for its status today, it may be an example of that phenomenon whereby the first widely popular version of something becomes frozen as a common term or cultural icon. Even in the 1990s, a comic-strip bathtub very likely sits off the floor on claw feet.

… it is the first thing that leaps to mind, the thing that is easy to recognize, the thing that is easy to doodle on a napkin, when one thinks generically or popularly about sort routines.

Non-archaelogical analysis

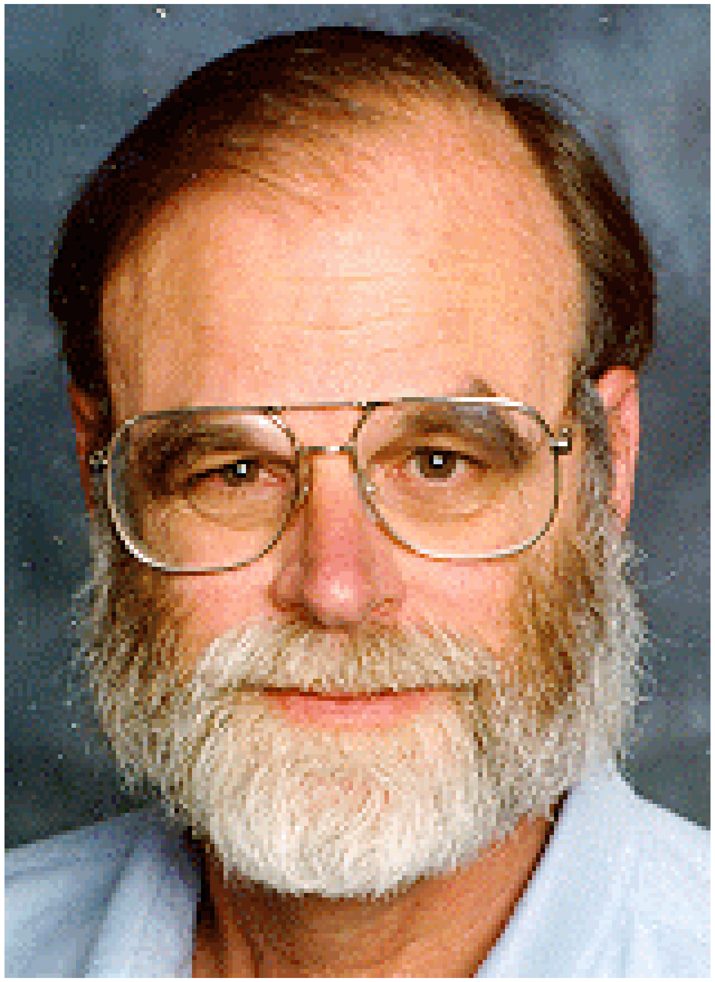
- Frank Bates and Mary Douglas
  - Prentice-Hall 1967, *Programming Language One*
  - Answer to “sort the array exercise” in book
- What is the complexity of this sort (substitute N for 10)



```
SORT:          /* SORT THE NUMBERS INTO ASCENDING ORDER */
               DO I = 2 BY 1 TO 10 ;
                  IF A(I-1)>A(I) THEN
                     DO ;
                        /* TRANSPOSE THE NUMBERS */
                        TEMP = A(I) ;
                        A(I) = A(I-1) ;
                        A(I-1) = TEMP ;
                        GO TO SORT ;
                     END ;
               END ;
```

Jim Gray (Turing 1998)

- Bubble sort is a good argument for analyzing algorithm performance. It is a perfectly correct algorithm. But it's performance is among the worst imaginable. *So, it crisply shows the difference between correct algorithms and good algorithms.*

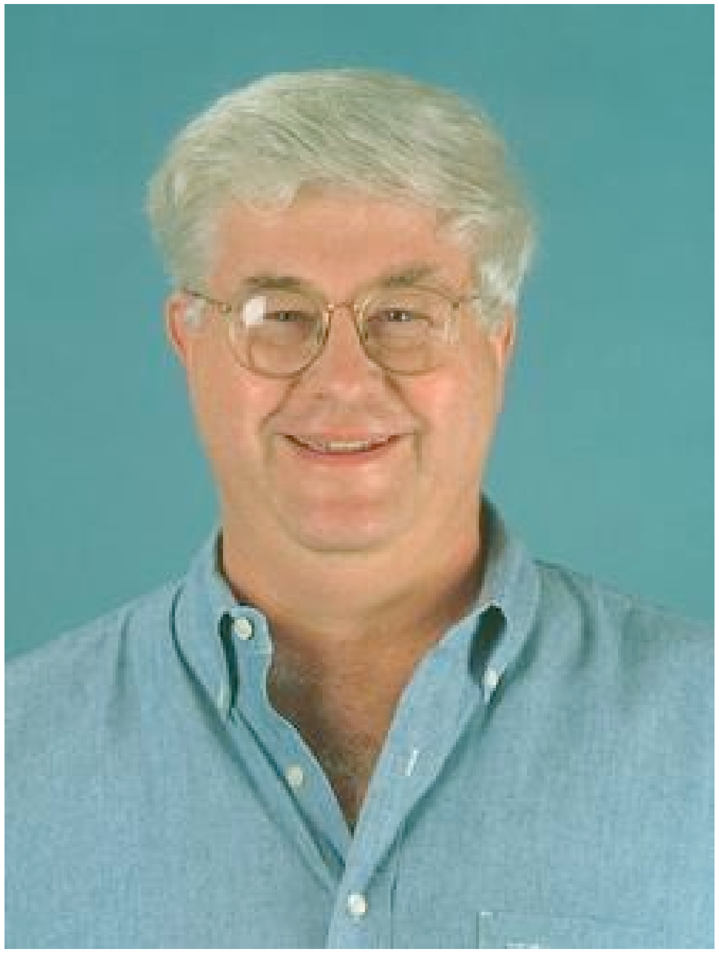


(italics mine)

Brian Reid (Hopper Award 1982)

Feah. I love bubble sort, and I grow weary of people who have nothing better to do than to preach about it. Universities are good places to keep such people, so that they don't scare the general public.

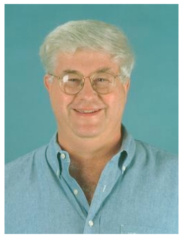
(continued)



Brian Reid (Hopper 1982)

I am quite capable of squaring N with or without a calculator, and I know how long my sorts will bubble. I can type every form of bubble sort into a text editor from memory. If I am writing some quick code and I need a sort quick, as opposed to a quick sort, I just type in the bubble sort as if it were a statement. I'm done with it before I could look up the data type of the third argument to the quicksort library.

I have a dual-processor 1.2 GHz Powermac and it sneers at your N squared for most interesting values of N. And my source code is smaller than yours.



Brian Reid
who keeps all of his bubbles sorted anyhow.

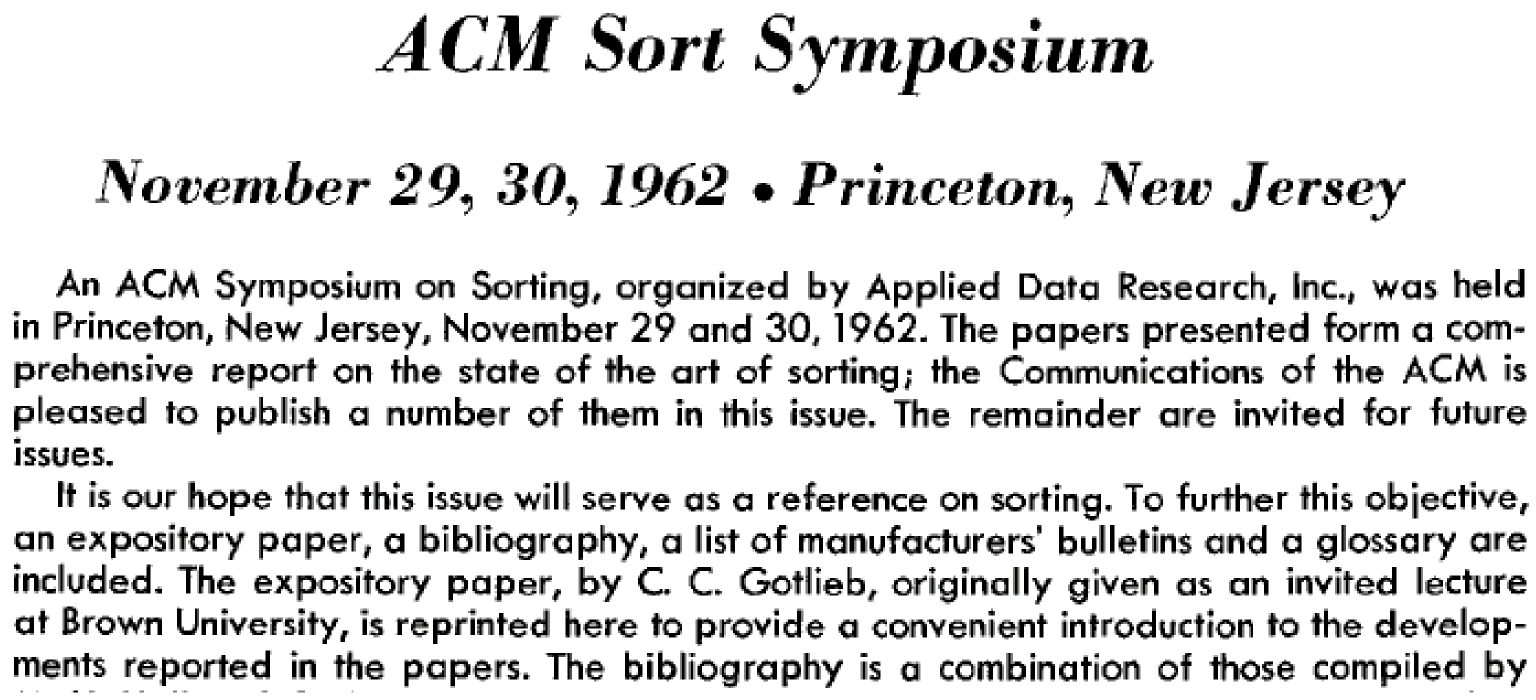
Good paper, by the way. Well written and actually has something to say.

A Bubble by any other name…

- **Official ACM version is shuttle sort (accepted wisdom, but…)**
- **Using Knuth and 1962 Sorting conference as a start, early papers and books all use "exchange sort" as the name**
  - **Demuth doesn't recall knowing sort as "bubble"**
- **1962: Iverson's *A Programming Language* uses "bubble sort"**
- **Subsequent works using "bubble sort" reference Iverson**

Reasonable evidence that Iverson's work is first occurrence of bubble sort in print --- not conclusive, but Occam's razor?

Sorting State of the Art circa 1962



ACM Sort Symposium

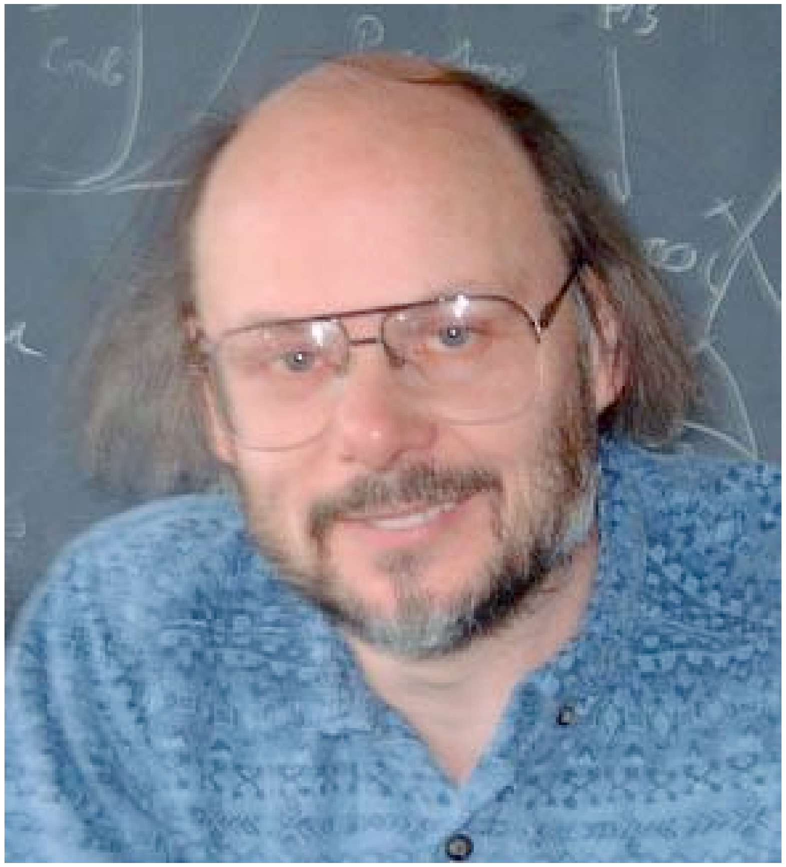
November 29, 30, 1962 • Princeton, New Jersey

An ACM Symposium on Sorting, organized by Applied Data Research, Inc., was held in Princeton, New Jersey, November 29 and 30, 1962. The papers presented form a comprehensive report on the state of the art of sorting; the Communications of the ACM is pleased to publish a number of them in this issue. The remainder are invited for future issues.

It is our hope that this issue will serve as a reference on sorting. To further this objective, an expository paper, a bibliography, a list of manufacturers' bulletins and a glossary are included. The expository paper, by C. C. Gotlieb, originally given as an invited lecture at Brown University, is reprinted here to provide a convenient introduction to the developments reported in the papers. The bibliography is a combination of those compiled by

Bjarne Stroustrup (Hopper Award 1993)

The only use of Bubble sort that I can think of is as "before" in a "before and after" study. It can be a good example of what not to do. I regularly get questions from relatively novice programmers who worry about "which sort algorithm to use" when they would be perfectly happy with the performance and interface of the standard library C++ sort() for problems 100 times larger and far more complex than what they have. Worse, they couldn't write a better sort if they tried.

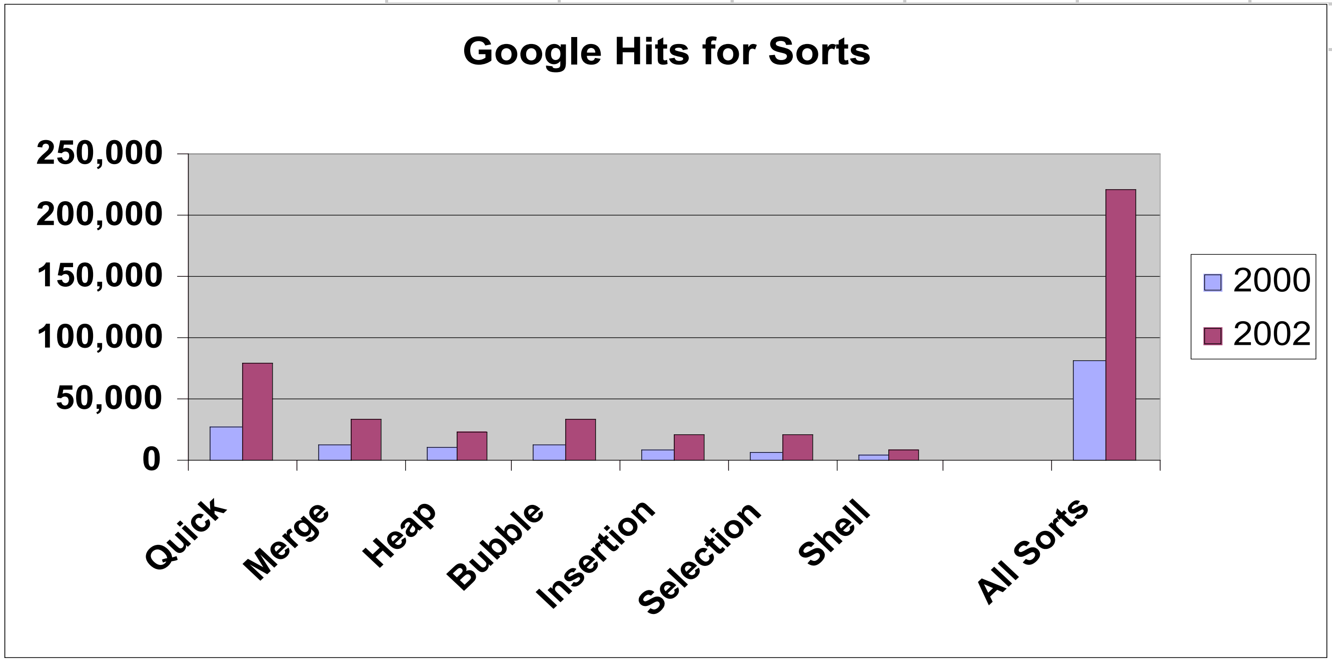


Bjarne Stroustrup (Hopper 1993)

Maybe (some) early programming teaching fail in giving students the impression that they need to write their own sorts? Another thing that people sometimes forget is that if you ever so infrequently sort 100 elements in a typical program, it doesn't really matter much which algorithm you use: unless your comparison function is horrendously slow, the average programmer wouldn't have a clue how to measure the minute time difference.

Popularity?

	2000	2002	percent growth
Quick	26,780	80,200	199.48%
Merge	13,330	33,500	151.31%
Heap	9,830	22,960	133.57%
Bubble	12,400	33,800	172.58%
Insertion	8,450	21,870	158.82%
Selection	6,720	20,600	206.55%
Shell	4,540	8,620	89.87%
All Sorts	82,050	221,550	170.02%



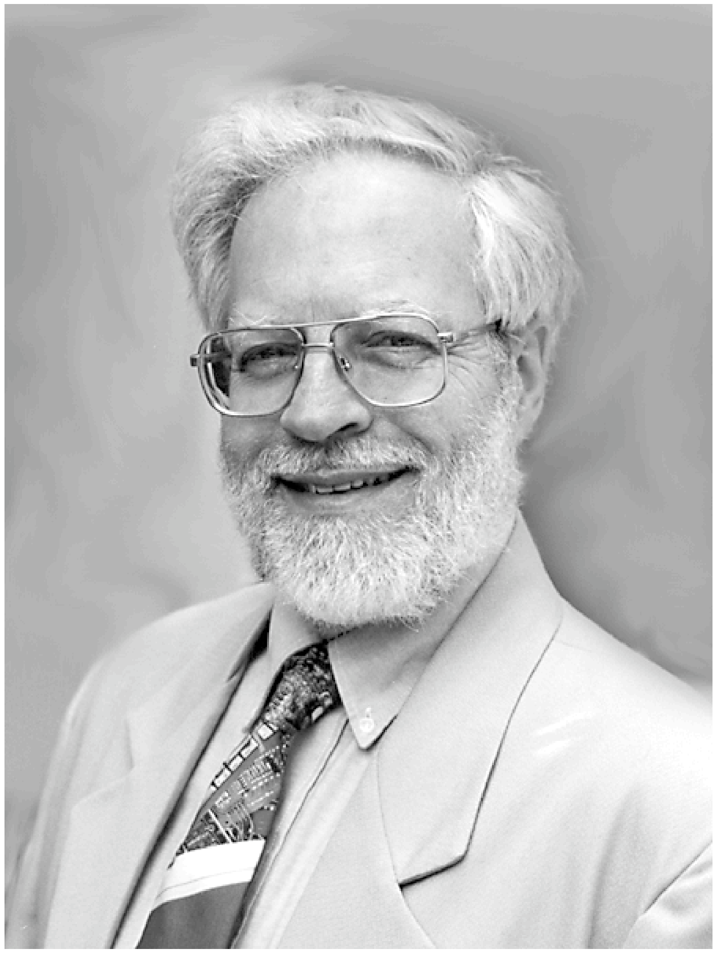
Dennis Ritchie (Turing Award 1983)

… in the mention of K&P Software Tools, you might observe that the apparent purpose of mentioning bubble first is to knock it down, since the n^2 behavior is stressed right away.

David Gries (Karlstrom '95, SIGCSE '91)

Hello! Nice to hear from you.

I never teach bubble sort. I see no reason to teach it. There are a lot more important things that I think students should see. When it comes to initial discussions of sorting, I teach selection sort and insertion sort.



Kind words for Bubble sort?

- However, the bubble sort is easy to remember and to program, and little time is required to complete a single step.

  Sorting, ACM Computing Surveys, 1971

- However, we assume that the number of items to be sorted is moderately large. If one is going to sort only a handful of items, a simple strategy such as the $O(n^2)$ "bubble sort" is far more expedient.

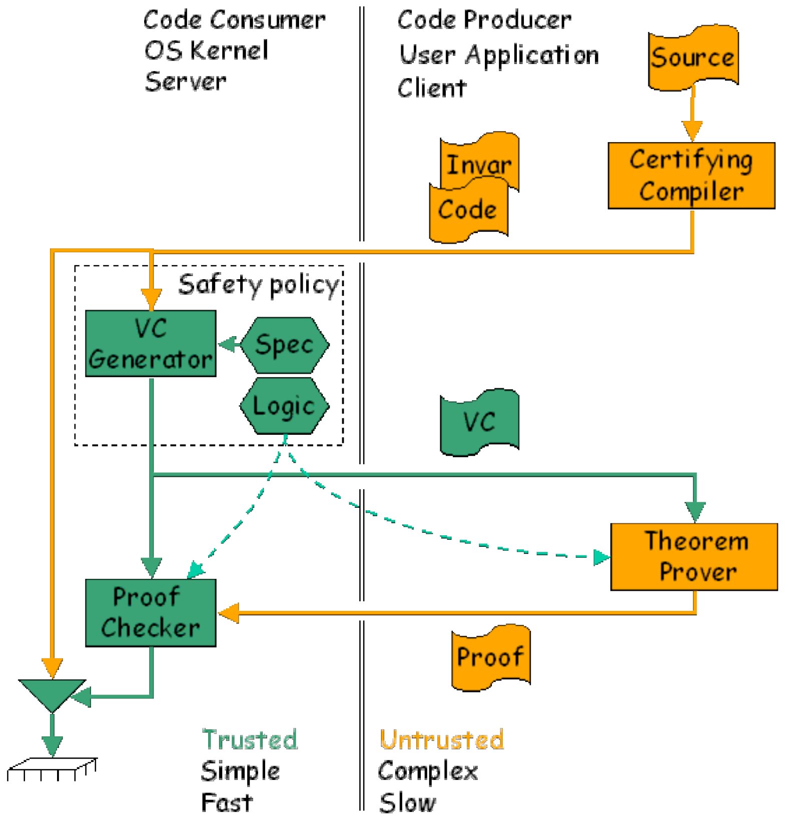
  Aho, Hopcroft, Ullman, *Algorithms*, 1974

- Nevertheless, this sorting algorithm is commonly used where the value of n is not too large and programming effort is to be kept to a minimum. … The bubble sort has the additional virtue that it requires almost no space in addition to that used to contain the input vector.

  Stanat and Mcallister, *Discrete Math for CS*, 1977

George Necula (Hopper Award 2001)

I am not sure what kind of comments you are looking for. I myself, do not like bubble sort and I prefer insertion sort, because it is easier to explain what the invariant is and why it is preserved at each step. I do think there is room in the curriculum for teaching O(n^2) sorting algorithms, because they are simple. But when it comes time to use them in practice why would anyone implement any sorting with the convenient availability of quicksort in all respectable libraries?



Nell Dale (Karlstrom '01, SIGCSE '96)

Your discussion mentions and discards the premise that bubble sort is valuable because it is O(n) when the data is sorted or almost sorted, because insertion sort is better under these conditions. My comment relates to this property. .... Letting students see that an algorithm can be made "smarter" is a valuable lesson. So give bubble sort its due: it makes other sorts look better and it gives the students a lesson in algorithms.

What do we teach in CS1?

- Take only memories, leave only …
  - What memories?
- Copyright term extension act
  - Wrong course?
- Is there a place for "bad" algorithms?
  - Exemplars of best practice?

- Three things in my course:
  1. `1 + 2 + … + N = N(N+1)/2`
  2. $2^{10} = 1024$
  3. The answer to most question is "it depends"

Peter Denning (Karlstrom ’96, SIGCSE ’99)

I first heard the term bubble sort when I was a grad student at MIT, approximately 1964. We considered it to be the same as the exchange sort, but with an onomatopoetic name to help visualize the process by which it worked. ...

The moral was that the conceptually simplest or most obvious algorithm is not always the most efficient; in fact, your quick and dirty solution might be too slow to be useful.

Dan McCracken SIGCSE '92, CPSR '89

In Programming Business Computers (1959, with Harold Weiss and Tsai Hwa Lee) we described it, but called it exchange sorting. I think the analysis is correct. If Big-Oh notation was in use for such things at the time I certainly hadn't heard of it, but we said the average time was n(n-1)/2.

My 1957 Digital Computer Programming has nothing on sorting.

Robert Aiken (SIGCSE ’95)

Like most of my CS colleagues I would include it [bubble sort] in a collection of sorting methods as I showed why, though it is easy to understand, it is not efficient. I think that is an important point and one students should learn early - both for historical as well as pedagogical reasons.

Thanks for your note. I quickly perused your paper and found it well-written and enjoyable. Nice job!

Essentially “No Comment”

- John McCarthy
  - “Sorry, I can’t recall anything that would be helpful.”
- Maurice Wilkes
  - “I am sorry that I cannot help you with your query about bubble sort. Obviously, you will be consulting Donald Knuth.”
- Richard Stearns
  - “No comment.”
- David Harel
  - “Thanks for considering me, but I don’t have anything of interest on this”
- Andy Tanenbaum
  - “… I am not really a sorting guru”

Knuth on Bubble sort (from The Art…)

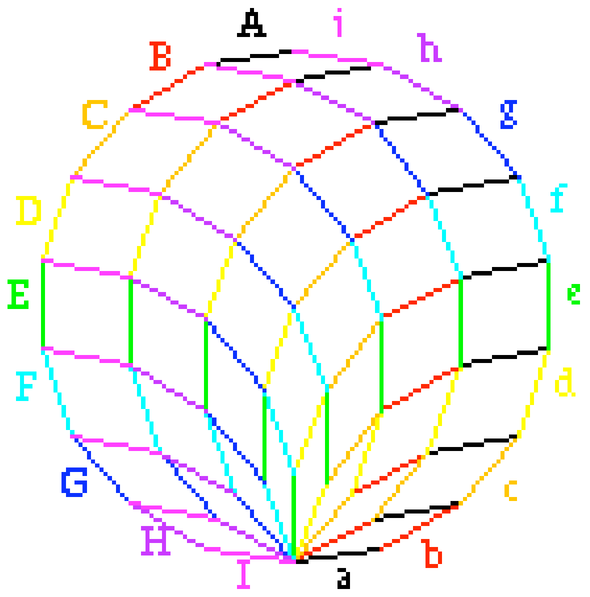
"In short, the bubble sort seems to have nothing to recommend it, except a catchy name and the fact that it leads to some interesting theoretical problems."

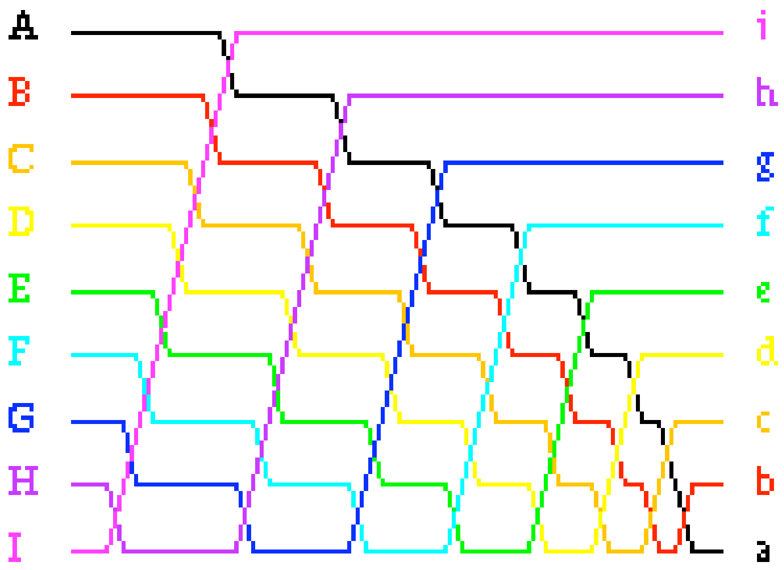
The Art of Computer Programming: Vol. 3, Sorting and Searching

"From a mathematical standpoint, Demuth's thesis was a beautiful piece of work. … But from a practical standpoint, the thesis was of no help. In fact, one of Demuth's main results was that in a certain sense "bubble sorting" is the optimum way to sort. … It turns out that [all other sorting methods studied] are always better in practice, in spite of the fact that Demuth has proved the optimality of bubble sorting on a certain peculiar type of machine."

The Dangers of Computer Science Theory

Crystalline structures = 2^{10} words


A
i
h
B
g
C
f
D
E
e
F
d
G
c
H
b
I
a


A
B
C
D
E
F
G
H
I
i
h
g
f
e
d
c
b
a