At execution of $A[f] := r$; on line prior to that labeled *schell*, f has not necessarily been assigned a value. f has a value if, and only if, the Boolean expression $B[k] > 0 \wedge B[k] < B[m]$ is **true** for at least one of the relevant values of k. In particular when matrix A is set up by $A[i] := i$; for each i the Boolean expression above is **false** on the first call.

ALGORITHM 202 is the best and fastest algorithm of the exicographic set so far published.

A collected comparison of these algorithms is given in Table I. t_n is the time for complete generation of $n!$ permutations. Times are scaled relative to t_8 for Algorithm 202, which is set at 100. Tests were made on an ICT 1905 computer. The actual time t_8 for Algorithm 202 on this machine *was* 100 seconds. r_n has the usual definition $r_n = t_n/(n \cdot t_{n-1})$.

TABLE I

Algorithm	t_6	t_7	t_8	r_6	r_7	r_8
87	118	—	—	—	—	—
102	2.1	15.5	135	1.03	1.08	1.1
130	—	—	—	—	—	—
202	1.7	12.4	100	1.00	1.00	1.00

CERTIFICATION OF:

ALGORITHM 258 [H]
TRANSPORT
[G. Bayer, *Comm. ACM 8* (June 1965), 381]
ALGORITHM 293 [H]
TRANSPORTATION PROBLEM
[G. Bayer, *Comm. ACM 9* (Dec. 1966), 869]

LEE S. SIMS (Recd. 21 Feb. 1967 and 17 Mar. 1967)
Kates, Peat, Marwick & Co., Toronto, Ont., Canada

Both of these algorithms were coded in Extended ALGOL 60 and tested on a Burroughs B5500. Three problems were solved correctly, one of them being of medium size (55 × 167). On this larger problem *transp*1 was found to be about twice as fast as *transport*.

In coding and debugging *transp*1 three apparent errors were found. In the right-hand column on page 870, after line 27 which is

i := *listu*[*u*]; *nlvi* := *nlv*[*i*];

a line is missing. This line should read

for *s* := (*i*−1) × *n* + 1 **step** 1 **until** *nlvi* **do**

Also in the right-hand column, the line

*s*4: ;

should be inserted ahead of line −12, which begins

comment Step 4. A column *j* with *b*[*j*] has been labeled, *b*[*j*]

On page 871, in the left-hand column, line −22 which reads

for *s* := 1 **step** 1 **until** *n* **do**

should read

for *s* := *l* **step** 1 **until** *n* **do**

CERTIFICATION OF ALGORITHM 285 [H]
THE MUTUAL PRIMAL-DUAL METHOD
[Thomas J. Aird, *Comm. ACM 9* (May 1966), 326]

H. SPÄTH (Recd. 13 Feb. 1967)
Institut für Neutronenphysik und Reaktortechnik, Kernforschungszentrum, Karlsruhe, Germany

The procedure *Linearprogram* has been translated into FORTRAN II and successfully run on the IBM 7074 Computer. The following corrections had been made (the first two are merely typographical errors).

1. P. 328, left column, 1 line after label *B*3:
reads:
if *A*[*row*[*k*−1, *i*],*col*[*k*, 0]] > **then**
should read:
if *A*[*row*[*k*−1, *i*],*col*[*k*, 0]] > 0 **then**

2. P. 328, left column, 1 line after label *B*4:
reads:
if *A*[*row*[*k*−1, *i*],*col*[*k*, 0]] > **then**
should read:
if *A*[*row*[*k*−1, *i*],*col*[*k*, 0]] > 0 **then**

3. P. 328, right column, after the end of the procedure *pickapivot* and before the label *NEXTPIVOT* there must be inserted the statement
col[0, 0] := 0;
Otherwise *col*[0, 0] has no assigned value when the procedure *subschema* is entered for the first time.

REMARK ON ALGORITHM 301 [S20]
AIRY FUNCTION [Gillian Bond and M.L.V. Pitteway, *Comm. ACM 10* (May 1967), 291]

M.L.V. Pitteway (Recd. 19 May 1967)
Brunel University, ACTON, W.3., England

The initial minus sign has been omitted from the line immediately following the line

end calculation of derivatives;

The statement should read

$p := -(rtmdx/xi) \times (2 \times A[2] + 4 \times A[4] + 6 \times A[6] + 8 \times A[8] + 10 \times A[10]);$

ACM will be pleased to replace any copy of the Communications of the ACM that has been damaged in the mail. Please return the damaged copy to ACM Headquarters in New York with your request.