T or U would result in an expression involving the dominant
term $n^2(m - n)$. Clearly, algorithms T and U are more time
efficient for $m$ sufficiently larger than $n$. They also enable
computation of only part of the remainder.

The above conclusions have been substantiated by the results of
computer programs implementing the three algorithms.

The programs were run on a CDC CYBER-73 computer. Table 1
shows results pertaining to the division of a poly-
nomial of degree 50 by polynomials of varying degree $n$. In the
table $U$ stands for the algorithm as specified, and $U^*$ stands for
a version based on equation (8). Except for $m$ and $n$, all entries
are in milliseconds.

References
BROWN, W. S. (1971). On Euclid's algorithm and the computation of
greatest common divisors, JACM, Vol. 18, pp. 478-504.
No. 4, July 1964, pp. 1547-1562.

Book reviews

Applications of Algol 68, Conference proceedings edited by V. J.
Rayward-Smith, 1976; 264 pages. (University of East Anglia,
£6-50)

"Which of languages W, X, Y, . . . , is best suited to my problem?"

is the first question that a multi-lingual programmer should ask when
starting a new project. This book contains evidence on how well or
ill suited ALGOL 68 (or more often ALGOL 68R) has already
shown itself to be in a dozen or more projects in fields as diverse as
automatic text editing, compiler writing, polynomial manipulation,
a mailing list system, interactive graphics, and initial programming
courses. Other papers—there are 26 in all—deal with more general
experience such as use of subsets, introduction of high level macros,
and the results of some benchmark tests.

Most of the myths are shown up for what they are—wishful
thinking generated as counter-propaganda by competitors in the
rat race. Thus at the Pierre et Marie Curie University in Paris, use of
ALGOL 68 in an initial course in computer design and shorter debugging times than
in previous years; the only 'undesirable' result was that as the good
became better the gap between good and bad students widened.
Sceptical comment (= sceptical?, or septic?) from colleagues showed
itself to be based on fears that shallow understanding of the nature of
computing would be shown up. Of particular interest is the bench-
mark report; untuned programs in ALGOL 60, FORTRAN
and ALGOL 68 ran at much the same speed; each language had its own
style of 'tuning for speed' and ALGOL 68 could (admittedly at some
trouble) be more finely tuned than the others. It then reached a speed
hardly distinguishable from Pascal, a language in which facilities
have been denied to the user in order to force the runtime pace.
But where there is criticism, 68R is often preferred, an unexpected
conclusion which seems to have a moral for compiler writers, to wit,
that if you need facilities to debug your compiler which run counter
to your high level philosophy (e.g. procedure bodies in machine
code), these should be left in as 'additional, possibly machine-
dependent' facilities, because sooner or later some user will also
need them, and if you withdraw them from the compiler before releasing
it, he will reject your implementation.

B. HIGMAN (Lancaster)

Press, £2.25)

This volume is one of the International Tracts in Computer Science
and Technology and their Application. The General Editors were
N. Metropolis, E. Piore and S. Ulam. The administrative editors
were Mark I. Halpern and William C. McGee. The contributing
editors were Louis Bollett, Andrei P. Ershov, and J. P. Laskin
(however, none of these appear to have contributed). The contents are:
A Tutorial on Data-Base Organisation, R. W. Engles; General
Concepts of the Simula 67 Programming Language, J. D. Ishihara
and S. P. Morse; Incremental Compilation and Conversational
Interpretation, M. Berthaud and M. Griffiths; Dynamic Syntax: A
Concept for the Definition of the Syntax of Programming Languages,
K. V. Hanford and C. B. Jones; An Introduction to ALGOL 68,
H. Bekic; A General Purpose Conversational System for Graphical
Programming, O. Lecarme; Automatic Theorem Proving Based on
Resolution, A. Pirrotte; A Survey of Extensible Programming
Languages, N. Solntseff and A. Yezerski.

This is a collection of tutorial and survey articles, with a sprinkling
of novel ideas; they are of a kind and length which are not readily
published elsewhere; and their publication in an occasional review is
to be welcomed. The article on Automatic Theorem Proving by
Resolution is particularly clear, though the method is not now
considered as promising as it once was.

C. A. R. HOARE (Belfast)

The Computer Journal