WILKINSON, J. H. (1960). "Rounding Errors in Algebraic Processes," *Information Processing*, pp. 44–53. Paris: UNESCO; Munich: Oldenbourg; London: Butterworths.
WOODS, L. C. (1953). "The Relaxation Treatment of Singular Points in Poisson's Equation," *Quart. J. Mech.*, Vol. 6, p. 163.
YOUNG, D. (1954). "Iterative Methods for Solving Partial Differential Equations of Elliptic Type," *Trans. Amer. Math. Soc.* Vol. 76, p. 92.

Note on a method of forming a sorting key for a partly ordered list, and an application

By D. M. Collison

In commercial applications, calculating the keys for items to be sorted is usually trivial, since they are rarely time-dependent. In on-line applications the keys may well vary with time, and if a complicated decision is involved in calculating them, any method of simplifying the decision is useful. One way may be to assume the answer, and check the assumption in a simpler manner after the calculation. For a key with two different forms, A and B, the conventional structure of the sorting routine is shown in Fig. 1, and the suggested one in Fig. 2. The new routine behaves like a flip-flop, the decisions switching it from one state to the other; and it can be extended for keys with more than two different forms, although the result is not as efficient. The main advantage of such a structure is that it will work better if the list is already partly ordered, as the cross-overs are less efficient than the conventional structure.

In the application, it was necessary to sort items with Cartesian coordinates (x, y) into angular order. Forming the polar coordinate Θ was slow and inefficient, as the key was only needed for the sorting, and the following key f was used (the case $x = y = 0$ could not occur).

$x \geqslant y \geqslant 0$	$f = y/8x$	$-\frac{1}{2}$
$y \geqslant \lvert x \rvert$	$f = -x/8y$	$-\frac{1}{4}$
$-x \geqslant \lvert y \rvert$	$f = y/8x$	
$-y \geqslant \lvert x \rvert$	$f = -x/8y$	$+\frac{1}{4}$
$x \geqslant -y \geqslant 0$	$f = y/8x$	$+\frac{1}{2}$

f is continuous over the interval $[-\frac{1}{2}, +\frac{1}{2}]$ and any two keys can be compared by one subtraction. The key for $y = 0$, $x > 0$ should be calculated by only one of the first or last forms, and not by both.

The routine was implemented on the Elliott 502. This computer has no modulus facility and the decision whether $|y| > |x|$ takes about six instructions. Form A, therefore, divides y by x and tests for overflow (one instruction), while form B divides x by y. The actual routine to deal with the five different forms is rather more complicated than Fig. 2.

Two points arose in the implementation and are included to guide possible users. The first occurs if the hardware division process does not round properly and the full length of the quotient is used. Adjacent items may be sorted into the wrong order. Luckily the 502 forms an extra digit for the quotient and rounds accordingly; with other computers it may be necessary to round and truncate the quotient by program. Another possibility is that both x/y and y/x may cause overflow. This occurs on the 502 if $x = y$, but it was possible to make provision for it without slowing down the two main loops of the routine.

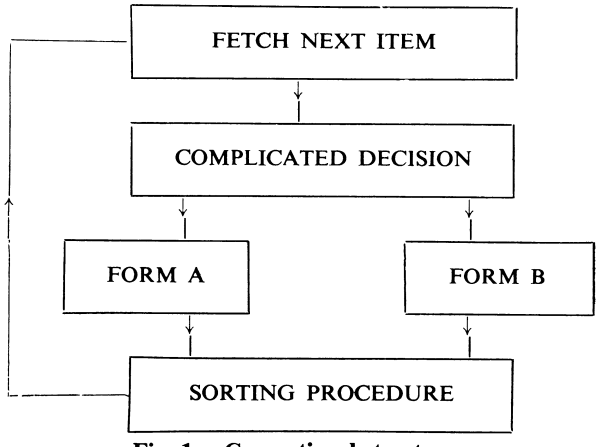

Fig. 1.—Conventional structure

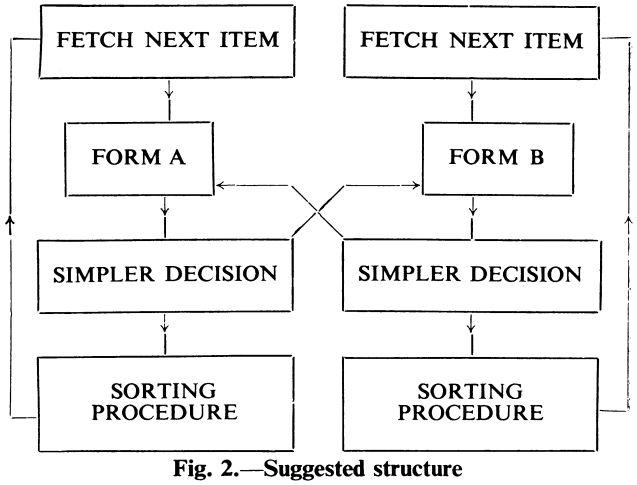

Fig. 2.—Suggested structure

The author would like to thank Elliott Brothers (London) Ltd. for permission to publish this note; and Mr. E. M. Shorter, who implemented the routine on the 502.