SOR23: A program to sort the input data for Step 2/3
by S. Söderhjelm

1. Introduction

The output from the Step 1 processing consists of a star-wise list of abscissae (with their standard deviation) for each set. The step 2/3 processing needs instead all observations of a given star, and thus a sort has to be performed. Because the set-results are already (or can be) ordered, a rather simple merge-sort can be used. One simply takes the first observation from each set and selects the lowest star-number for output. A new observation from this set is input, and a new lowest star-number is output. By this principle, the output will consist of the observations in the required star-wise order.

2. The merge-sort

The input to the actual merge is a direct access file with LBUF observations per record and with five integers per observation: SETNO, STARNO, IABSC1, IABSC2, ISDABSC: (The d.p. abscissae values have been converted to a suitable 2-integer format, as outlined e.g. by C. Petersen 840925). The buffer-length parameter LBUF should be as large as possible while allowing an LBUF x 5 x KWAY array to be stored in primary memory. Here KWAY is the smallest power of two such that NSET ≤ KWAY, thus for realistic set-numbers one will have KWAY = 2048. On the present HP3000-configuration at Lund Observatory, the corresponding limit is LBUF = 10.

(Originally, I envisaged a multi-pass program with KWAY << NSET. One would then sort consecutive batches of KWAY sets each. Then, in further passes, KWAY batches would be sorted, then batches of batches, and so on. After introducing the "selection-tree" described below, the single-pass mode was found to be superior.)

The sort-algorithm consists mainly in choosing the lowest star-number among NSET (~ 2000) observations. The first version of the program used NSET-1 direct comparisons for each observation processed, but to save computing time one may preferably use a "selection-tree" (see e.g. Horowitz and Sahni, Fundamentals of Data Structures, Pitman, 1977). In this way, only \( \log_2 \text{KWAY} \) (~ 12) comparisons are needed to update the selection-tree, and its "root" node points to the next observation to be output.

Superficially then, the program works by reading the first LBUF observations from each set into the primary memory array IARR(5, KWAY+1, LBUF). The selection-tree indicates the "winning" set, and this observation is transferred to an output buffer. The next observation from this set is already in IARR, and the process is rapidly repeated until all LBUF observations from any set are used up. A new buffer is then input from the observation-file, and the merging continues. When the star-number in the output-buffer changes, it is output to a sequential output-file which will contain thus (variable length) records of the form:

\[
\text{STARNO, NOBS, (SETNO, IABSC1, IABSC2, ISDABSC)} \ldots \text{repeated NOBS times} \ldots
\]

This is the required input to Step 2/3.
3. Practical details

The input for the present program is the integer-tape from SIP23 as described in NDAC/L0/048. The first step is then to transfer these data to the direct-access disc-file described above. Especially in the multi-pass tests with small KWAY, it was apparent that I/O-operations make a dominating contribution to the computing times. Therefore, there is some gain in using system read/write routines and files instead of Fortran ones. The actual sort then creates a sequential output that may be either left as a Fortran-file on disk or else written out directly on tape.

The tape-output is preferrable when disk-space has to be saved, and from the tests performed it is not markedly slower. A full 2000-set sort (723000 observations of 20000 stars) took less than 3 hours from tape to tape, while a tape to disk run with 730 sets still took more than an hour. (A version with system write to the final disk-file has not been tested due to the large disk-space needed with the simple fixed-length format used for the tape; at present 1200 bytes/star).

4. Conclusions

The present program can be easily adopted to the real task of sorting all the Step 1-results (100 000 stars, 2000 sets) to star-wise order. The time needed would be of the order of 12 hours, depending on the disk- and tape-hardware available. The minimum disk-space needed (tape to tape mode) is about 72 Kb.