ESTIMATING LGP-30 RUN-TIME

Business Applications Dept.
ROYAL McBEE CORPORATION

#TBL-5

Data Processing Applications

It is desirable to be able to make an estimate of machine running time without making a detailed layout of the computer program. It is believed that in most instances the following estimating standards will yield a result accurate to within ± 20% of the overall time required. If the problem is of a very unusual nature, or the time must be determined with greater exactitude, dummy programming should be written, the number of instructions per program step estimated, and time estimated on the basis of 3/4 drum revolution per instruction. This latter procedure can only be performed by a computer applications analyst.

The following estimating standards are expressed in words per minute. A "word" is defined as a code word or piece of data; e.g., a customer code, a clock number, a pay rate, a quantity, an amount, or what have you. This approach has been chosen in lieu of an approach based on individual characters because experience has shown that over a range of code words and data for most problems, the number of characters per word averages out very consistently to between five and six, and it is obviously much less work to count 500 words than to count 3000 characters. Since the standards are based on 6 characters per word, they may, if desired, be very easily converted to characters per second by dividing by ten. (6 char/word ÷ 60 secs/min).

The standards are divided in six sections:

1. Tape punching
2. Input
3. Output
4. Storage
5. Computation
6. Off-line Print

ESTIMATING STANDARDS

		NOTES
A. Tape Punching:		
10-Key Adding Machine:	25 words/min.	
Full Keyboard Adding Machine	20 words/min.	
Typewriter *	40 words/min.	(1)
B. Input:		
342 Reader - Hexadecimal:	750 words/min.	(7A)
342 Reader - Decimal:	250 words/min.	(2) (7A)
Typewriter (either):	75 words/min.	(7A)
C. Output:		
Typewriter decimal print:	75 words/min.	(3) (7) (7B)
Typewriter decimal punch:	75 words/min.	(3)
342 decimal punch:	125 words/min.	(3)
D. Storage:		
342 Hexadecimal read-in:	750 words/min.	(4)
342 Hexadecimal punch-out:	150 words/min.	(4)

E. Computation:

15% of (input + output + Storage) time (5)

Plus additions for:

1) Memory or table search: (5A)

$T - .0015 NL$

where: T = additional time in minutes
N = number of searches
L = locations per search

2) Sorting of numbers: (5B)

$T = .00055 SI^2$

where: T = additional time in minutes
S = No. of sets to be sorted
I = No. of items per set

F. Off-line Print: (6)(7)(7B)

Typewriter: 80 words/min.

* Plain English only - not "keypunching"

NOTES

(1) Tape Punching - the typewriter speed is based on a mixture of numeric data and alphabetic words, as in a billing operation. Straight numeric data would be considerably slower (about 17 words/minute or so).

(2) Input - the photoreader decimal input speed includes the time required to test the code words, construct an address and binarize and store the data. If desirable from a sales standpoint, input speed can arbitrarily be stated as 1250 to 1500 words/minute, and the difference in residual time included under the "computation" section.

(3) Output - the output time also includes various computation and test time, and can, if desired, be stated at up to 100 words/minute for the typewriter and 200 words/minute for the 342 punch, with the difference in time being included in computation.

(4) Storage - machine language (hexadecimal) punching and reading are covered under this section rather than under input and output, because in a functional sense a machine language tape (e.g.., inventory balance, sales-to-date, etc.) punched out for subsequent re-entry into the machine constitutes an extension of memory.

(5) Computation - the computation time allowance for internal housekeeping and transfer of data, plus normal amounts of computation to make extensions, derive percentages and ratios, optimize results, etc., is established as 15% of total input, output and storage time. In applications where an exceptionally large amount of computation is involved (e.g., a complex incentive payroll calculation), this allowance should be raised to 20% to be on the safe side. This basic allowance is not sufficient to cover certain computer operations which involve highly repetitive internal manipulation. Prime among such operations are searching for information (as in looking across the drum for an inventory code word) and sorting (as in internally arranging data in code-word

sequence). If such operations cannot be avoided, additional computation time should be allowed as follows:

A. Memory or table search:

$T = .0015\ NL$
where: T = additional time in minutes
N = number of searches
L = locations per search

e.g., 100 code words and 100 data words are to be read into memory. Each code word requires a search of one track in memory looking for an old balance for that item. Extra time allowance -

$N = 100$
$L = 64$
$T = .0015 \times 100 \times 64 = 9.6$ mins.

B. Sorting of numbers:

$T - .00055\ SI^2$
where: T = additional time in minutes
S = number of sets to be sorted
I = number of items per set

e.g., a daily time card for each of 100 men is to be punched on tape and read into memory. Each man averages 5 jobs per day, in random order. Time cards are in clock number order. LGP-30 is to sort the five (avg.) jobs for each man into job number order so that the result is:

```
Clock # 1
  Job    1
  Job    3
  Job    4
  etc.

Clock # 2
  Job    1
  Job    2
  Job    5
  etc.
```

B. (Continued)

Extra time allowance:

$S = 100$
$I = 5$
$T = .00055 \times 100 \times 5 \times 5 = 1.4$ mins.

Note that if each man worked on 20 jobs, Time would go up steeply:

$T = .00055 \times 100 \times 20 \times 20 = 22$ mins.

Certain pseudo-sorting routines exist whose time vs. number of-items-sorted characteristics are markedly more linear than those shown above. It is difficult to generalize about these routines and so no estimating standards for them are included here.

(6) Off-line Print - faster than on-line print because it does not include output computation.

(7) In figuring the number of printed output words, one word should be added for each line of print-out, and one-half word added for each column on each line (e.g., a 1000 word report, made up of 10 columns across the page and 100 lines down the page would equal 1000 words of data plus 100 words for carriage returns plus $10 \times 100 \times \frac{1}{2} = 500$ words for spacing between columns, or a total of 1600 words). This means that the blank spaces on a report cost almost as much time as do the printed characters. As a result, output format should be as condensed as possible.

In cases where either the input data words or the print-out words seem to average a good deal smaller or larger than five or six characters, they may be counted and a character-per-second approach used. On this basis, the following rules should be observed:

A. Input - use the average number of characters per word including both code words and data. Do not count signs and stop codes.

B. Output - use the largest number of characters which will appear in each column, plus sign, plus number of spaces between columns. Add 5 characters for each carriage return.

(8) In cases where only part of the data on a decimal input tape is used on a particular run (as in an inventory problem where the same transaction tape is processed against 4 drumloads of balances) Hexadecimal reading speeds may be applied to those words not used in each run, e.g.; If a tape is read 4 times, it can be figured as being read once at 250 WPM, and 3 times at 750 WPM.

EXAMPLE

Problem

Inventory control - Read in a hex tape containing balances and minimums on 1500 inventory items. Read in part number and quantity withdrawn for 300 inventory transactions. Update 150 affected. Print out part number, balances, and minimum for 50 items below minimum. Punch out updating tape for 150 changed balances.

Time Estimate

Input:

342 D/R: $\frac{\text{300 items} + \text{300 code words}}{\text{250 words/min.}}$ = 2.4 mins.

Output:

```
50 items x 3 words/item          =   150 words
50 lines x 3 columns/line x 1/2  =    75 words
50 lines                         =    50 words
                     Total Words     275

   Time = 275 ÷ 75 words/min. = 3.7 mins.
```

Storage

```
342 read = 1500 items x 2 words/item ÷ 750 words/min.
             = 4.0 mins.
342 punch = 150 balances + 150 addresses = 300 words
             300 words ÷ 120 words/min.   = 2.4 mins.

   Total Storage Time - 6.4 mins.
```

Computation:

```
     Input     =    2.4 min
     Output    =    3.7 min
     Storage   =    6.4 min
                   12.5 min
               x    15%
 Computation   =    1.88 min
```

Total Run-Time

```
               1 - O - S   =  12.5 min
             Computation   =   1.9 min.
      Total Machine Time   =  14.4 min
```

Tape Preparation

10-key add-punch: 600 words ÷ 25 words/min. = 24 mins.