

HIGH SPEED SORT AND SEARCH

THIS PROGRAM IS RESTRICTED TO
MEMBERS OF THE ORGANIZATION
TO NON-MEMBERS IS PROHIBITED

I. FUNCTION

1. To sort a group of words or blocks of words into an area of the drum. That is, to store each word or block of words in the sorting area so that all the designating keys will be in ascending order. As each new word or block is stored, the previously stored words or blocks are moved aside to make room, when ever necessary.
2. To search an area of the drum containing sorted words or blocks of words to find the address of the particular word that matches the desired key.

II. METHOD OF OPERATION

1. Sorting is accomplished by originally clearing all the words in the sorting area to zero. Then, as each new word or block is brought to the sorting program to be stored in the sorting area, the key is used to calculate a probable address within the area. If the address is still zero, the word or block will be stored there. If there is already a word or block stored at the probable address, the routine advances or backs through the sorting area looking for an open (zero) memory location. If the word or block should be inserted between two other words already stored with no zero location between them, the program shifts the stored words up or down to make room until a zero location is found. (See Appendix 'A' for flow diagram and formula derivation.)
2. Searching is accomplished by the same address calculation method as sorting. The program goes directly to the probable address, and then searches up or down until it finds desired key or passes the spot it should be. The routine will exit with the address of the desired key in the accumulator at a Q of 29. If the key is not found, the program (as written) will stop.

III. DEFINITIONS

The following terms are described to establish their meaning as they apply to the initializing, sorting and searching routines.

1. Block Size -- A block is one unit record; that is, one message. It may be only one LGP-30 word long or it may be more than one. The block size is the number of LGP-30 words in the block at a Q of 30.
2. Key -- The key is the word or portion of a word which designates a particular block. It must always be the first word in a multi-word block. It may be any number, positive or negative, or alphanumeric representation, that can be contained in an LGP-30 word. It may be at any Q; all keys within one sorted group or area, however, must be at the same Q. Since the key may not completely fill the first (or only) word of a block, a mask is used to extract it from the word. In the case of positive number keys, the key may be located anywhere in the word. In the case of negative number keys, the key, and its mask, must be located in the most significant bits of the LGP-30 word, and include the sign bit. If the entire word is used as the key, the mask would be 'wwwwwwq'. If 'zero' is to be a possible key, the

High Speed Sort & Search

entire word cannot be used for the key; a bit must be stored someplace behind the mask to enable the program to recognize the 'zero' as a valid key word, and not one of the unused parts of the sorting area.

3. Limits -- The portion of the program that calculated the probably address must know the general range of the keys to be sorted or searched for. The program will sort or search for keys outside the range, but an approximate upper and lower limit must be known. For instance, in sorting performance percentages, it might be known that almost all will fall between 50% and 150%. However, a percentage as high as 800% or as low as 0% might possibly be encountered. For this application, the upper limit should be 150 and the lower 50. This would yield much greater sorting speed than using 800 and zero. If it were not known what numbers would be encountered, the limits would have to be the largest and smallest numbers that could be held at the Q of the key.

Limits are always at the same Q as the keys. Limits are absolute numbers. In the case of negative numbers, the lower limit is the smallest absolute (most negative) number.

4. Sorting Area -- The area of the drum into which the words or blocks will be sorted or stored should be larger than that required to contain the file. Less shifting and searching, and faster operating speeds result from a larger sorting area. For example, 100 words will be sorted into 128 locations much faster than into 100 locations. A restriction of this fact is that any routine which prints out the contents of the sorting area after sorting must contain steps to test for, and skip over, the zero locations. If more blocks or words are read into the sorting area than it can contain, the program will hang up in an endless loop looking for an open location. (This can be prevented by programming a counter as part of the input routine, one for each sorting area.) The sorting area is defined by a start and an end address at a q of 29. The start is numerically smaller than the end. The maximum size of the sorting area is governed only by computer capacity and the need for other programs.
5. Multi-word Blocks -- Blocks may be as long as desired. Each additional word in a block, however, adds to the time required to shift or store that block in the sorting area. To sort, the program is entered with the first word of the block in the accumulator (see the calling sequence). The balance of the block must be previously stored at the end of the sorting routine (starting at $L_0 + 0501$). These are temporary storage locations and will be cleared to zero by the sorting routine.

The block will end up stored in the sorting area with the key word in the first (low order) word and the balance of the block in the immediately following words of the sorting area. For example, the first or key word could be a station number and the next four words could be temperatures. After sorting, they might end up with the key in 2140 and the temperatures in 2141-2144.

IV. USE OF PROGRAM

The program has four calling sequences; two initializing, one for sorting and one for searching. The following rules govern their use:

LGP-30 USERS' ORGANIZATION - POOL

High Speed Sort & Search

1. Prior to sorting or searching, the program must have been initialized at some time.
2. Prior to sorting the first or a new batch of data, the sorting area must be cleared to zero (initialize and clear).
3. If any of the six constants (start, end, block size, upper limit, lower limit, and mask) change, the program must be reinitialized.
4. Sorting and searching may be done continuously and alternately except as noted in 1, 2 and 3 above.

Below is a description of how the program might be used to set up and use two tables of constants.

Setting Up

1. Initialize and clear sorting area # 1
2. Read in and sort constants into area #1
3. Initialize and clear sorting area # 2
4. Read in and sort constants into area #2

Searching

5. Initialize for area #1
6. Search for data in area #1
7. Search for more data in area #1
8. Initialize for area #2
9. Search for data in area #2

Adding

10. Read in and sort new constant into area #2
11. Initialize for area #1
12. Read in and sort new constants into area #1
13. Search for data in area #1

An area must be cleared to zero before data can be sorted into it. Searches can be made for that data and more data sorted into the area without re-initializing. However, if the area is to be used for a new batch of data, the initializing and clearing steps must be executed to clear out the old data. If the program is to be used for searching and sorting in more than one area, the initializing steps must be executed before each transfer from one area to the next.

High Speed Sort & SearchV. CALLING SEQUENCES

1. Initialize and Clear

<u>Location</u>	<u>Inst.</u>	<u>Address</u>	<u>Remarks</u>
Q	R	L ₀ + 0332	
Q + 1	U	L ₀ + 0446	
Q + 2	XZ	L (start)	At q of 29
Q + 3	XZ	L (end)	At q of 29
Q + 4	XZ	Block Size	At q of 29
Q + 5		(Lower Limit)	Same Q as key
Q + 6		(Upper Limit)	Same Q as key
Q + 7		(mask)	To extract key
Q + 8		Etc.	

2. Initialize

<u>Location</u>	<u>Inst.</u>	<u>Address</u>	<u>Remarks</u>
X	R	L ₀ + 0332	
Q + 1	U	L ₀ + 0313	
Q + 2	XZ	L (start)	At Q of 29
Q + 3	XZ	L (end)	At Q of 29
Q + 4	XZ	Block Size	At q of 29
Q + 5		(Lower Limit)	Same Q as key
Q + 6		(Upper Limit)	Same Q as key
Q + 7		(Mask)	To extract key
Q + 8		Etc.	

3. Sorting

<u>Location</u>	<u>Inst.</u>	<u>Address</u>	<u>Remarks</u>
Q - 1	B	L (key word)	
Q	R	L ₀ + 0244	
Q + 1	U	L ₀ + 0110	
Q + 2		Etc.	

4. Searching

<u>Location</u>	<u>Inst.</u>	<u>Address</u>	<u>Remarks</u>
Q - 1	B	L (key word)	
Q	R	L ₀ + 0145	
Q + 1	U	L ₀ + 0031	
Q + 2		Etc.	

Note: Q - 1 in the Sorting and Searching need not contain a B order; any order or orders which leaves the key word in the accumulator is permissible.

VI. STORAGE

The program requires 320 locations of instructions and constants (5 tracks). If the blocks of data are more than one word long, the additional words must be

High Speed Sort & SearchVI. STORAGE (Continued)

stored in the locations following the program starting in ($L_0 + 0501$).

VII. TIME:

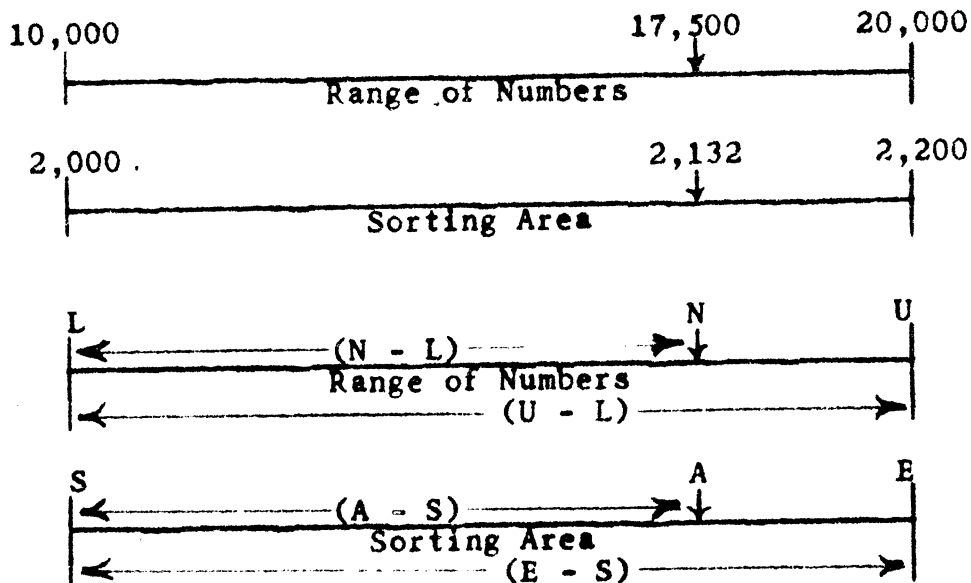
Initializing - One second
 Initializing and Clearing - One second plus three seconds
 per track of sorting area.
 Searching - Variable, 1/2 second minimum
 Sorting - Variable, 1/2 second minimum

Examples of Sorting Rates:

<u>Block Size</u>	<u>Sorting Area Size</u>	<u>File Size</u>	<u>Approximate Time to Sort</u>
1	128	100	45 - 65 sec.
1	1280	1000	450 - 650 sec.
2	256	100	75 - 105 sec.

VIII. ALTERNATE NOT IN FILE SEARCH

If it is desired to continue with a program when searching, and the key is not in file, the address of the next step should be inserted in 'T' step $L_0 + 143$ in place of T 0320.

LGP-30 SORTING ROUTINEDERIVATION OF CALCULATION FORMULA

L = Lower limit of numbers
 U = Upper " " "
 S = Address of start of sorting area
 E = " " end " " "
 N = Number to be sorted
 A = Current address
 B = Multiplying factor
 C = Addition factor

$$\frac{(A - S)}{(N - L)} = \frac{(E - S)}{(U - L)}$$

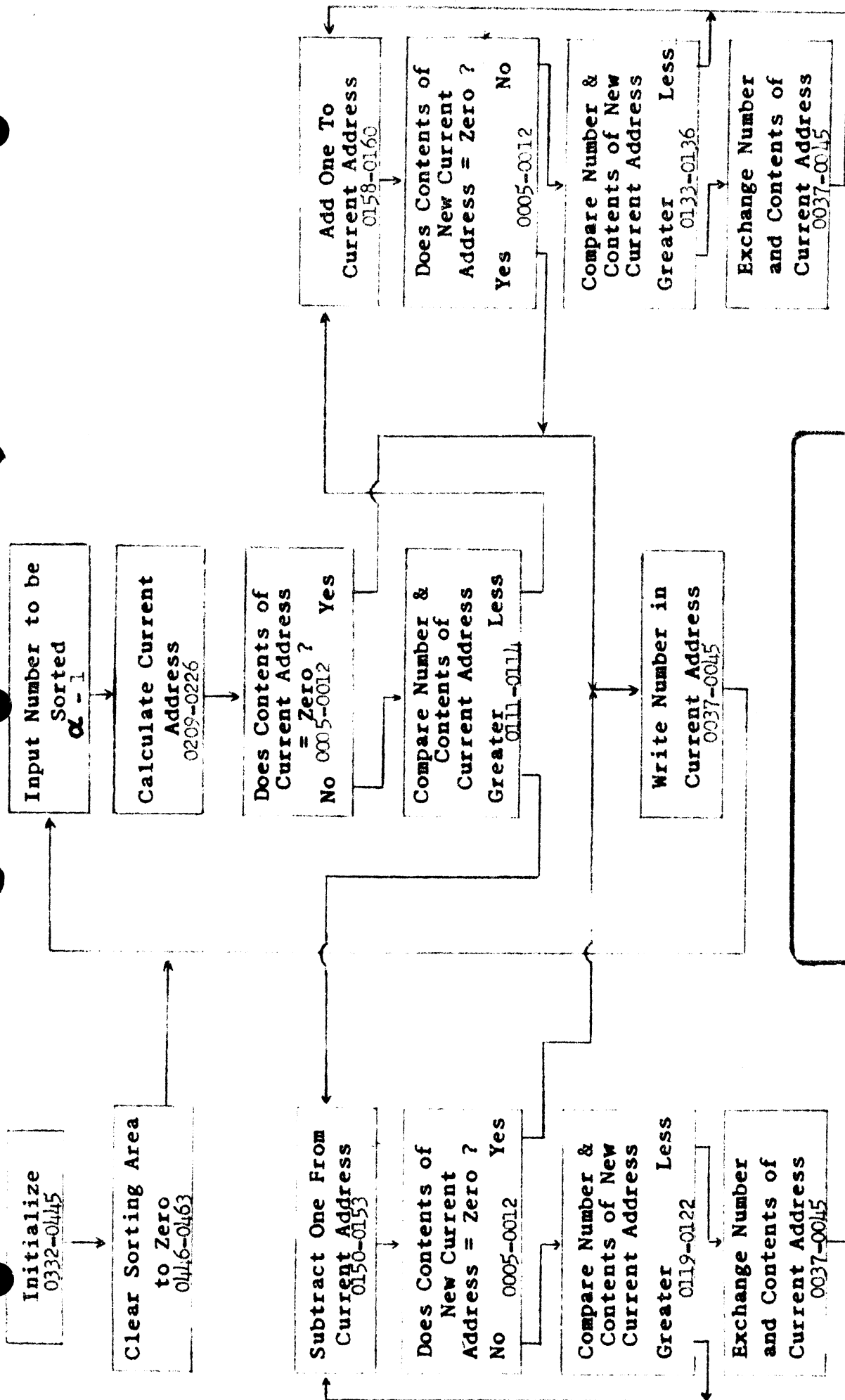
$$A - S = \left(\frac{E - S}{U - L} \right) (N - L)$$

$$A - S = N \left(\frac{E - S}{U - L} \right) - L \left(\frac{E - S}{U - L} \right)$$

$$A = N \left(\frac{E - S}{U - L} \right) + \left[S - L \left(\frac{E - S}{U - L} \right) \right]$$

$$A = N \times B + C; \text{ where } B = \left(\frac{E - S}{U - L} \right), \text{ and } C = \left[S - L \left(\frac{E - S}{U - L} \right) \right]$$

(Note: B and C are calculated during initializing portion of routine.)



LGP-30 USERS' ORGANIZATION - POOL
LGP-30 SORTING ROUTINE
 "High Speed Sort & Search"
 Program No. LL-92



PREPARED FOR: LGP-30 USER'S ORGANIZATION - POOL				PAGE 1 OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C.S. Kreger	PROGRAM CHECKED BY:	DATE Sep. 9, 1959
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK

PROGRAM INPUT CODES	LOC	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/					(0=Non Optimum Address)	
	/	<input checked="" type="checkbox"/>					(0434)(0214)
		0 0 0 0	Z	0 0 0 0	/	(E-S)	Multiply Factor <input checked="" type="checkbox"/>
Current		0 1	A	0 0 2 3	/	(U-L)	Z[E+1]
Address		0 2	Y	0 0 3 8	/		Current Address
Testing		0 3	S	0 1 4 6	/	<input checked="" type="checkbox"/>	Z[Start]
		0 4	T	0 2 0 0	/		→ (Search) not in file
		0 5	B	[0 0 0 0]	/		→ (Sort) up or error
		0 6	U	0 0 0 7	/		Word from sorting area
Testing		0 7	H	0 0 5 7	/	<input checked="" type="checkbox"/>	T.W.
For		0 8	S	0 2 3 0	/		1@30
Zero		0 9	T	0 1 2 4	/		→ Neg. or zero word
Word		1 0	A	0 0 5 3	/		1@30
		1 1	E	0 0 6 1	/	<input checked="" type="checkbox"/>	Mask
		1 2	C	0 0 6 2	/		T.I.
		1 3	U	0 0 6 0	/		→ To switch address (0203)(0249)
		1 4	XZ	0 0 4 9	/		constant (0220)
0 0 0 0 0 0 1		1 5		W W Q	/	<input checked="" type="checkbox"/>	Mask
		1 5	Z	[]	/		Block Size (0357)(0158)
		1 7	R	0 0 6 0	/		<input checked="" type="checkbox"/>
Searching		1 8	U	0 1 5 0	/		→ Mod. ↑ or down
Down		1 9	S	0 0 6 2	/	<input checked="" type="checkbox"/>	T.I.
(A)		2 0	A	0 0 6 3	/		S.I.
Section		2 1	T	0 1 5 0	/		→ Not found-repeat
		2 2	U	0 1 4 7	/		→ B section
		2 3	Z	0 0 0 0	/	<input checked="" type="checkbox"/>	[E+1] (0353)(0001) <input checked="" type="checkbox"/>
		2 4	R	0 0 6 0	/		
Search		2 5	U	0 1 5 8	/		→ Mod. adr. up
Up		2 6	B	0 0 6 2	/		T.I.
(A)		2 7	S	0 0 6 3	/	<input checked="" type="checkbox"/>	S.I.
Section		2 8	T	0 1 5 8	/		→ Not found-repeat
		2 9	U	0 1 3 9	/		→ B section
		3 0	B	[]	/		B[start] (0348)(0104) <input checked="" type="checkbox"/>
Search U		3 7	R	0 0 6 0	/	<input checked="" type="checkbox"/>	

PREPARED FOR: LGP-30 USERS' ORGANIZATION - POOL			PAGE 2	OF 10
JOB NO.	PROGRAM NO. 1 LL-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	
PROBLEM: HIGH SPEED SORT AND SEARCH			DATE 9/9/59	
			TRACK	

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	<input checked="" type="checkbox"/>					
SEARCH		0,0312	u	0,209	/	-->	compute
FIRST		3 3	b	0,062	/		T.I.
TEST		3 4	s	0,063	/		S. I.
A		3 5	t	0,024	/	<input checked="" type="checkbox"/>	--> set up search up
SECTION		3 6	u	0,046	/	-->	B Section
		3 7	b	0,059	/		W.W.
SORT		3 8	h	[]	/		sort area (0002) current adr
EXCHANGE		3 9	u	0,040	/	<input checked="" type="checkbox"/>	
AND		4 10	b	0,062	/		T.I.
WRITING		4 11	h	0,063	/		W.I.
OF		4 12	b	0,057	/		T.W.
KEYS		4 13	u	[]	/	<input checked="" type="checkbox"/>	(0044) --> mult word block 0403
		4 14	h	0,059	/		WW
		4 15	u	[]	/		0158 0150 up-down 0130 0116 0202 write (0200)
SEARCH		4 16	c	0,318	/		dump
FIRST		4 17	s	0,062	/	<input checked="" type="checkbox"/>	TI
TEST		4 18	a	0,063	/		S.I.
B		4 19	t	0,017	/	-->	set up search down
SECTION		5 10	u	0,144	/	-->	FOUND
		5 11	u	0,216	/	<input checked="" type="checkbox"/>	Inst. Const. (0419)
		5 12	b	0,501	/		" " (0237)
0 0 0 0 0 1,1		5 13		2	/	1 30	(0010)(0124)
		5 14	[]	[]	/		(end)(0106)(0351)
		5 15	[]	[]	/	<input checked="" type="checkbox"/>	1 10 block size (0361) (0219)
		5 16	[]	[]	/		E+1 (0354) (0277) (0054)
		5 17	[]	[]	/		test word
		5 18	[]	[]	/	n	mult. factor (0429)(0215)
		5 19	[]	[]	/	<input checked="" type="checkbox"/>	search word writing word
		6 10	f	[]	/		
		6 11	[]	[]	/		Mask (0415) (0011)
		6 12	[]	[]	/		Test Ident. (key)
		6 13	[]	[]	/	<input checked="" type="checkbox"/>	search Ident. writing Ident. (key)

PREPARED FOR: LGP-30 USERS' ORGANIZATION - POOL			PAGE 3 OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:
PROBLEM: HIGH SPEED SORT AND SEARCH			DATE 9/9/59
			TRACK

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		T.O.S.	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	<input checked="" type="checkbox"/>					
BLOCK		0,1,0,10	h	0,0,0	/		Input
EXCH			u	0,2,5,2	/		--> Block Each
			z	0,0,0,0	/		Add Factor (0439)(0216)
FIRST			b	0,0,0,5	/	<input checked="" type="checkbox"/>	Current Adr.
SORTING OR			s	0,0,3,0	/		B[start]
SEARCHING			t	0,1,0,8	/		
ADDRESS			b	0,0,5,4	/		[End]
OUTSIDE			u	0,2,2,6	/	<input checked="" type="checkbox"/>	
AREA			b	0,1,4,6	/		[start]
			u	0,2,2,6	/		
sortlu			r	0,0,6,0	/		
			u	0,2,0,9	/	<input checked="" type="checkbox"/>	--> Compute adr.
			b	0,0,6,2	/		T.I.
			s	0,0,6,3	/		W.I.
			t	0,1,2,9	/		--> set up sort up
			b	0,1,1,8	/	<input checked="" type="checkbox"/>	[0150] Down adr mod
			y	0,0,4,5	/		Exch. Exit
			r	0,0,6,0	/		
			u	0,1,5,0	/		--> Mod adr down
			s	0,0,6,2	/	<input checked="" type="checkbox"/>	T.I.
			a	0,0,6,3	/		W.I.
			t	0,1,5,0	/		--> Repeat testing down
			u	0,0,3,7	/		--> exchange
			c	[] [] [] []	/	<input checked="" type="checkbox"/>	Holder (0452)(0458)
			a	0,0,5,3	/	1 30	
ZERO			t	0,0,1,1	/		--> neg key word
TEST			u	0,2,4,5	/		--> zero word
			h	[] [] [] []	/	<input checked="" type="checkbox"/>	H(E+1) (0155)
			x z	0,0,0,1	/	1 29	(0456)
			b	0,1,3,5	/	z(0158)	up mod. adr.
			y	0,0,0,5	/		exch. switch
			r	0,0,6,0	/	<input checked="" type="checkbox"/>	

LGP-30 CODING SHEET

PREPARED FOR: LGP-30 USERS' ORGANIZATION - POOL				PAGE 4 OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	DATE 9/9/59
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	3 12	u	0 1 5 8	/	--> Mod. adr up	
SORT		3 12					
OR		3 13	b	0 0 6 2	/	T.I.	
TESTING		3 14	s	0 0 6 3	/	W.I.	
		3 15	t	0 1 5 8	/	<input checked="" type="checkbox"/> --> Repeat testing up	
		3 16	u	0 0 3 7	/	--> Exch	
		3 17	u	0 0 3 3	/	Address (0201)	
		3 18	z	0 0 0 0	/	Dump (0323) (0252)	
		3 19	c	0 3 1 8	/	<input checked="" type="checkbox"/> dump	
SEARCH		4 10	s	0 0 6 2	/	T.I.	
OR		4 11	a	0 0 6 3	/	S.I.	
BI		4 12	u	0 1 4 3	/		
SECTION		4 13	t	0 3 2 0	/	<input checked="" type="checkbox"/> --> Not In File	
SEARCH		4 14	b	0 0 3 8	/	FOUND bring our print address	
BI TEST		4 15	u	[]	/	--> SEARCH EXIT	
SEARCH	R	4 16	z	0 0 0 0	/	START	
		4 17	b	0 0 6 2	/	<input checked="" type="checkbox"/> T I	
SEARCH		4 18	s	0 0 6 3	/	S I	
DOWN B		4 19	u	0 1 4 3	/		
SECTION		5 10	c	0 2 0 8	/	DUMP	
MODIFY		5 11	s	0 0 1 6	/	<input checked="" type="checkbox"/> Z[] BLOCK Size	
ADDRESS		5 12	a	0 0 3 8	/	H[] Current address	
DOWN		5 13	u	0 1 5 4	/		
		5 14	y	0 0 0 5	/	Bring step	
ACURRENT		5 15	s	0 1 2 7	/	<input checked="" type="checkbox"/> H [E=1]	
ADDRESS		5 16	t	0 0 0 1	/	--> ok keep testing	
TESTING		5 17	u	0 1 6 2	/	--> not in file or down or error	
		5 18	b	0 0 1 6	/	z[] Block size	
MODIFY		5 19	a	0 0 3 8	/	<input checked="" type="checkbox"/> H [] current address	
ADDRESS		6 10	u	0 1 5 4	/		
UP		6 11	u	0 1 1 2	/	(0246) test factor (0201)	
		6 12	u	0 1 1 2	/	(0201)	
		6 13	y	0 0 3 8	/	<input checked="" type="checkbox"/> mask (0220)	

PREPARED FOR: LGP-30 USER'S ORGANIZATION- POOL				PAGE 6 OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C.S. Kreger	PROGRAM CHECKED BY:	DATE Sep. 9, 1959
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	<input checked="" type="checkbox"/>					
Block		0 2 3 12	Y	0 2 6 1	/		sort area B step
Exchange		3 3	B	0 2 5 8	/	0	input area B step
Address		3 4	S	0 2 6 3	/	1 @ 29	
Mod.		3 5	Y	0 1 0 0	/	<input checked="" type="checkbox"/>	input area H step
		3 6	Y	0 2 5 8	/		input area B step
		3 7	S	0 0 5 2	/	B[0501]	
		3 8	T	0 0 4 5	/		→ Block exch exit
		3 9	V	0 2 5 8	/	<input checked="" type="checkbox"/>	→ exch. steps
		4 10	Z	0 0 0 0	/		temp.stor. 1 (0225)(0218)
		4 11	X.Z	0 0 0 1	/	1 @ 29	(0255) (0205)
Set up		4 12	R	0 0 4 5	/	0	
Writing		4 13	U	0 0 3 7	/	<input checked="" type="checkbox"/>	
SortR: Exit		4 14	U	0 0 0 0	/		sorting exit
		4 15	B	0 0 6 0	/		
Zero		4 16	S	0 1 6 1	/	U[0112]	
Word		4 17	T	0 2 4 9	/	<input checked="" type="checkbox"/>	searching
Fpund		4 18	U	0 2 4 2	/		sorting -write
Separate		4 19	A	0 0 1 4	/	Z[0049]	
Routines		5 10	T	0 3 2 3	/		→ searching up or down
		5 11	U	0 1 5 8	/	<input checked="" type="checkbox"/>	→ 1st search add one to adr
		5 12	B	0 1 3 8	/		temp stor (0231)
Exchange		5 13	H	0 0 0 0	/	0	sorting area
Rest		5 14	B	0 2 6 1	/	0	sort area B step
of		5 15	S	0 2 4 1	/	<input checked="" type="checkbox"/> 1 @ 29	
words		5 16	U	0 2 3 1	/		
in		5 17	Z	0 0 0 0	/	@ 10	block size (0221)(0362) (0308)(0236)
multi-		5 18	B	[] []	/	0	input area word <input checked="" type="checkbox"/>
word		5 19	H	0 1 3 8	/	<input checked="" type="checkbox"/>	temp. stor.
block		6 10	U	0 2 6 1	/		(0254)(0232)
		6 11	B	[] []	/	0	sorting area word
		6 12	U	0 1 0 0	/		
		6 13	X.Z	0 0 0 1	/	<input checked="" type="checkbox"/> 1 @ 29	(0234)

LGP-30 CODING SHEET

PREPARED FOR: LGP-30 USER'S ORGANIZATION - POOL			PAGE 7	OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	
PROBLEM: HIGH SPEED SORT AND SEARCH			DATE Sep. 9, 1959	
			TRACK	

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	X					
		0,3,10,10	X	P 0,0,0,0	/	1 @ 12	(0360)
Initialize		10,11	N	0,2,3,0	/	1 @ 30	
prevent		10,12	H	0,0,5,9	/	0	(U-L)
divide		10,13	B	0,3,1,8	/	X	(E-S)
overflow		10,14	S	0,0,5,9	/		(U-L)
		10,15	U	0,3,2,8	/		
		10,16	B	0,3,2,1	/	Z(0500)	input area start
setup		10,17	A	0,3,2,2	/	X @ 29	block 12
block		10,18	Y	0,2,5,8	/		input area B step
exchange		10,19	B	0,0,3,8	/		current adr.
		11,10	A	0,3,2,5	/	@ 29	block size
		11,11	U	0,2,5,5	/	X	
Initialize		11,12	N	0,0,5,8	/		(0426)
	U	11,13	R	0,4,4,0	/		
setup		11,14	U	0,3,3,3	/	→	initialize
exit		11,15	U	0,3,3,2	/	X →	exit
block		11,16	H	0,0,5,9	/		
exchange		11,17	U	0,3,0,6	/		
		11,18	Z	0,0,0,0	/		dump (E-S)
		11,19	U	0,2,2,6	/	X	(0408)
		12,10	Z	0,3,1,6	/		[not stop in fill] (0402)
		12,11	Z	0,5,0,0	/		input area start (0306)
		12,12	Z	0,0,0,0	/	@ 29	block size
search		12,13	C	0,1,3,8	/	X	dump
testing		12,14	U	0,0,1,2	/		
		12,15	Z	0,0,0,0	/	@ 29	block size (0453)(0461)
		12,16	C	0,0,0,0	/		holder
		12,17	Z	0,0,4,4	/	X	(0406)
Initialize		12,18	T	0,4,3,2	/	→	divide will not overflow now
prevent		12,19	B	0,0,5,8	/		shift factor
divide		13,10	N	0,2,3,0	/	0 1 @ 30	
overflow		13,11	U	0,4,2,9	/	X	

PREPARED FOR: LGP-30 USER'S ORGANIZATION- POOL				PAGE 8	OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	DATE Sep. 9, 1959	
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK	

PROGRAM INPUT CODES	SOS	LOCATION	INSTRUCTION		SOS	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
Initialize	/						
Clear and Initialize	/	0,3,3,2	U	[]	/		(0346)
Initialize	R	3,3	B	0,3,3,2	/	0	$\alpha + 2$
		3,4	Y	0,3,4,7	/	0	
		3,5	A	0,2,6,3	/	\times 1 @ 29	
		3,6	Y	0,3,5,0	/	0	
		3,7	A	0,2,6,3	/	1 @ 29	
		3,8	Y	0,3,5,6	/	0	
		3,9	A	0,2,6,3	/	\times 1 @ 29	
		4,0	Y	0,4,1,0	/	0	
		4,1	A	0,2,6,3	/	1 @ 29	
		4,2	Y	0,4,1,2	/	0	
Initializing		4,3	A	0,2,6,3	/	\times 1 @ 29	
		4,4	Y	0,4,1,4	/	0	
		4,5	A	0,2,6,3	/	1 @ 29	
		4,6	Y	0,3,3,2	/	0	
		4,7	B	[]	/	\times	start 90334)
		4,8	Y	0,0,3,0	/	0	
		4,9	Y	0,1,4,6	/	0	
		5,0	B	0,0,0,0	/	0	end (0336)
		5,1	Y	0,0,5,4	/	\times 0	
		5,2	A	0,2,6,3	/	@ 29	
		5,3	Y	0,0,2,3	/	0	E+1
		5,4	Y	0,0,5,6	/	0	
		5,5	Y	0,1,2,7	/	\times	
		5,6	B	[]	/	@ 29	block size(0338)
		5,7	Y	0,3,2,2	/		
		5,8	Y	0,0,1,6	/		
		5,9	Y	0,3,2,5	/	\times 0	
		6,0	N	0,3,0,0	/	1 @ 12	
		6,1	H	0,0,5,5	/	0 @ 10	block size
		6,2	H	0,2,5,7	/	0	
		6,3	B	0,0,1,6	/	\times @ 29	block size

LGP-30 CODING SHEET

PREPARED FOR: LGP-30 USER'S ORGANIZATION - POOL				PAGE 9 OF 10
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	DATE Sep. 9, 1959
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	X					
		0,4,10,10	S	0,4,4,5	/	°2 @ 29	
		10,11	T	0,4,0,6	/	→	one work block
		10,12	B	0,3,2,0	/	°Z[0316]	
		10,13	Y	0,0,4,3	/	X	switch
		10,14	B	0,0,0,3	/	S[0146]	
		10,15	U	0,4,0,9	/		
		10,16	B	0,3,2,7	/	Z[0044]	
		10,17	Y	0,0,4,3	/	X	switch
		10,18	B	0,3,1,9	/	U[0226]	
		10,19	H	0,2,1,7	/	°	switch
		11,10	B	0,0,0,0	/	°	(L) (0340)
		11,11	H	0,0,5,7	/	X °	
Initializing		11,12	B	0,0,0,0	/	°	(U) (0342)
		11,13	H	0,1,3,8	/	°	
		11,14	B	0,0,0,0	/	°	(Mask)
		11,15	H	0,0,6,1	/	X °	
		11,16	B	0,1,3,8	/		(U)
		11,17	S	0,0,5,7	/		(L)
		11,18	H	0,0,5,9	/	°	(U-L)
		11,19	B	0,0,5,1	/	X U[0216]	
		12,10	H	0,2,1,5	/	°	switch
		12,11	B	0,0,5,4	/	°	end
		12,12	S	0,1,4,6	/	°	start
		12,13	H	0,3,1,8	/	X	(E-S)
		12,14	S	0,0,5,9	/	°	(U-L)
		12,15	T	0,4,3,2	/		divide will not overflow
Initialize		12,16	B	0,3,1,2	/	N[0058]	
prevent		12,17	H	0,2,1,5	/	X	switch
divide		12,18	B	0,0,5,3	/	1 @ 30	
overflow		12,19	H	0,0,5,8	/		shift factor
		13,10	B	0,0,5,9	/		(U-L)
		13,11	U	0,3,0,1	/	X	



PREPARED FOR: LGP-30 USER'S ORGANIZATION- POOL				PAGE 10 OF 10	
JOB NO.	PROGRAM NO. L1-92	PROGRAM PREPARED BY: C. S. Kreger	PROGRAM CHECKED BY:	DATE Sep. 9, 1959	
PROBLEM: HIGH SPEED SORT AND SEARCH				TRACK	

PROGRAM INPUT CODES	STOP	LOCATION	INSTRUCTION		STOP	CONTENTS OF ADDRESS	NOTES
			OPERATION	ADDRESS			
	/						
	/	0 4 3 2	B	0 3 1 8	/		(E-S)
		3 3	D	0 0 5 9	/	0	(U-L)
Initializing		3 4	H	0 0 0 0	/	(E-S) (U-L)	multiply factor
		3 5	M	0 0 5 7	/	X	(L)
		3 6	H	0 2 0 8	/		temp. stor.
		3 7	B	0 1 4 6	/		(S)
		3 8	S	0 2 0 8	/		temp. stor.
		3 9	H	0 1 0 2	/	X	add factor
		4 10	U	[]	/		exit or clear (0446)(0313)
		4 11	C	[]	/		holder (0462)(0450)
		4 12	C	0 5 7	/		dump
		4 13	C	[]	/	X	clearing step
		4 14	U	0 4 5 5	/		
		4 15	XZ	0 0 0 2	/	2 @ 29	(0205)(0400)
Initialize and		4 16	R	0 4 4 0	/		
clear		4 17	U	0 3 3 3	/	X	
		4 18	B	0 1 4 6	/		[start]
		4 19	Y	0 4 4 1	/		
Clear		5 10	Y	0 4 4 3	/		
sorting		5 11	B	0 0 2 3	/	X	[exit]
area		5 12	Y	0 1 2 3	/		
to		5 13	Y	0 3 2 6	/		
zero		5 14	U	0 4 4 2	/		
		5 15	B	0 4 4 1	/	X	
		5 16	A	0 1 2 8	/	1 @ 29	
		5 17	Y	0 4 4 3	/		
		5 18	S	0 1 2 3	/		
		5 19	T	0 4 6 1	/	X	
		6 10	U	0 3 3 2	/		→ not done
		6 11	A	0 3 2 6	/		done — exit
		6 12	C	0 4 4 1	/	C[E+1]	
		6 13	U	0 4 4 3	/	X	holder