PLANNING AND CODING OF PROBLEMS FOR AN ELECTRONIC COMPUTING INSTRUMENT

By
Herman H. Goldstine
John von Neumann

Part II, Volume II

PLANNING AND CODING OF PROBLEMS
FOR AN
ELECTRONIC COMPUTING INSTRUMENT

By

Herman H. Goldstine John von Neumann

Report on the Mathematical and Logical Aspects of an
Electronic Computing
Instrument

Part II, Volume II

The Institute for Advanced Study
Princeton, New Jersey
1948

PREFACE

This report was prepared in accordance with the terms of Contract No. W-36-034-ORD-7481 between the Research and Development Service, U. S. Army Ordnance Department and The Institute for Advanced Study. It is a continuation of our earlier report entitled, "Planning and Coding of Problems for an Electronic Computing Instrument", and it constitutes Volume II of Part II of the sequence of our reports on the electronic computer. Volume III of Part II will follow within a few months.

Herman H. Goldstine

John von Neumann

The Institute for Advanced Study
15 April 1948

TABLE OF CONTENTS

INTRODUCTORY REMARK

We found it convenient to make some minor changes in the positioning of the parts of all orders of Table II, and in the specific effect of two among them.

First: We determine that in each order of Table II the memory position number x occupies not the 12 left-hand digits (cf. the second remark of 7.2, cf. also 8.2), but the 12 right-hand digits. I.e. the digits 9 to 20 or 29 to 40 (from the left) for a left-hand or a right-hand order, respectively.

Second: We change the orders 18, 19 (the **partial** substitution orders xSp, xSp'), so that they substitute these new positions, and these crosswise. I.e. 18 replaces the 12 right-hand digits of the left-hand order located at x (i.e. the digits 9 to 20 [from the left]) by the 12 digits 29 to 40 (from the left) in the accumulator. Similarly 19 replaces the 12 right-hand digits of the right-hand order located at x (i.e. the digits 29 to 40 [from the left]) by the 12 digits 9 to 20 (from the left) in the accumulator.

For the standard form of a position mark, $x_0 = 2^{-19}x + 2^{-39}x$, as introduced in 8.2, these changes compensate each other and have no effect. Therefore all the uses that we made so far of these orders are unaffected.

On the other hand the new arrangement permits certain arithmetical uses of these orders, i.e. uses when the position x is not occupied by orders at all, but when it is used as (transient) storage for numbers. In this case the new arrangement provides very convenient 20-digit shifts, as well as certain other manipulations. These things will appear in more detail in the discussion immediately preceding the static coding in 10.7.

10.0 CODING OF SOME SIMPLE ANALYTICAL PROBLEMS

10.1 We will code in this chapter two problems which are typical constituents of analytical problems, and in which the approximation character of numerical procedures is particularly emphasized. This is the process of numerical integration and the process of interpolation, both for a tabulated one-variable function. Both problems allow a considerable number of variants, and we will, of course, make no attempt to exhaust these. We will nevertheless vary the formulations somewhat, and discuss an actual total of six specific problems.

10.2 We consider first the integration problem.

We assume that an integral $\int f(z)dz$ is wanted, where the function $f(z)$ is defined in the z-interval 0, 1, and assumes values within the range of our machine, i.e., size < 1. Actually $f(z)$ will be supposed to be given at N+1 equidistant places in the interval 0, 1, i.e., the values $f(h/N)$, $h = 0, 1, \ldots, N$, are tabulated. We wish to evaluate the integral by numerical integration formulae with error terms of the order of those in Simpson's formula, i.e., $O(1/N^4)$.

This is a method to derive such formulae. Consider the expression

$$(1) \qquad \varphi(u) = \int_{\frac{h-\xi u}{N}}^{\frac{h+\xi u}{N}} f(z)dz - \left[2f\left(\frac{h}{N}\right)\xi + \frac{1}{3}\left(f\left(\frac{h+u}{N}\right) - 2f\left(\frac{h}{N}\right) + f\left(\frac{h-u}{N}\right)\right)\xi^3\right]\frac{u}{N} .$$

There will be $h = 0, 1, \ldots, N$, $0 \leq \xi \leq 1$, $0 \leq u \leq 1$.

Simple calculations, which we need not give here explicitly, give $\varphi(0) = \varphi'(0) = \varphi''(0) = 0$ (indeed $\varphi'''(0) = 0$ and for $\xi = 1$ even $\varphi''''(0) = 0$), implying

$$(2) \qquad \varphi(u) = \int_0^u du_1 \int_0^{u_1} du_2 \int_0^{u_2} du_3\, \varphi'''(u_3),$$

and further

$$(3) \qquad \varphi'''(u) = \left[f''\left(\frac{h+\xi u}{N}\right) + f''\left(\frac{h-\xi u}{N}\right)\right]\frac{\xi^3}{N^3} - \left[f''\left(\frac{h+u}{N}\right) + f''\left(\frac{h-u}{N}\right)\right]\frac{\xi^3}{N^3} -$$

$$- \frac{1}{3}\left[f'''\left(\frac{h+u}{N}\right) - f'''(h-u)\right]\frac{\xi^3 u}{N^4} .$$

(3) transforms into

$$\varphi'''(u) = \left[f''\left(\frac{h+\xi u}{N}\right) - f''\left(\frac{h+u}{N}\right) + f'''\left(\frac{h+u}{N}\right)(1-\xi)\frac{u}{N}\right]\frac{\xi^3}{N^3} +$$

(4) $$+ \left[f''\left(\frac{h-\xi u}{N}\right) - f''\left(\frac{h-u}{N}\right) - f'''\left(\frac{h-u}{N}\right)(1-\xi)\frac{u}{N}\right]\frac{\xi^3}{N^3} -$$

$$- \left[f'''\left(\frac{h+u}{N}\right) - f'''\left(\frac{h-u}{N}\right)\right]\left(\frac{4}{3} - \xi\right)\frac{\xi^3 u}{N^4}$$

Putting

$$Mf'''' = \operatorname*{Max}_z |f''''(z)| \quad ,$$

we see that (4) yields

(5) $$|\varphi'''(u)| \leqq 2Mf''''\left(\frac{7}{3} - 2\xi\right)\frac{\xi^3 u^2}{N^5} \quad ,$$

and by (2)

(6) $$|\varphi(u)| \leqq \frac{Mf''''}{30}\left(\frac{7}{3} - 2\xi\right)\frac{\xi^3 u^5}{N^5} \quad .$$

Putting $u = 1$, and recalling (1), we find:

(7) $$\int_{\frac{h-\xi}{N}}^{\frac{h+\xi}{N}} f(z)dz = \left[2f\left(\frac{h}{N}\right)\xi + \frac{1}{3}\left(f\left(\frac{h+1}{N}\right) - 2f\left(\frac{h}{N}\right) + f\left(\frac{h-1}{N}\right)\right)\xi^3\right]\frac{1}{N} + \varphi,$$

$$|\varphi| \leqq \frac{Mf''''}{90}(7-6\xi)\ \xi^3\ \frac{1}{N^5} \ .$$

Putting $\xi = 1$ and summing over $h = \ell+1, \ell+3, \dots, k-1$ (ℓ, $k = 0, 1, \dots, N$, $k-\ell>0$ and even) we get Simpson's formula:

(8) $$\int_{\frac{\ell}{N}}^{\frac{k}{N}} f(z)dz = \sum_{h=\ell+1,\ell+3,\dots,k-1}\left[f\left(\frac{h+1}{N}\right) + 4f\left(\frac{h}{N}\right) + f\left(\frac{h-1}{N}\right)\right]\frac{1}{3N} + \varphi_1 =$$

$$= \left[\begin{matrix} f\left(\frac{\ell}{N}\right) + 4f\left(\frac{\ell+1}{N}\right) + 2f\left(\frac{\ell+2}{N}\right) + 4f\left(\frac{\ell+3}{N}\right) + \dots + \\ + 4f\left(\frac{k-3}{N}\right) + 2f\left(\frac{k-2}{N}\right) + 4f\left(\frac{k-1}{N}\right) + f\left(\frac{k}{N}\right)\end{matrix}\right]\frac{1}{3N} + \varphi_1 \ ,$$

$$|\varphi_1| \leqq \frac{Mf''''}{90}\ \frac{k-\ell}{2N^5} \leqq \frac{Mf''''}{180}\ \frac{1}{N^4} \quad .$$

Putting $\xi = \frac{1}{2}$ and summing over $h = \mathcal{X}+1, \mathcal{X}+2, \ldots, k-1$ ($\mathcal{X}$, $k = 0, 1, \ldots, N$, $k-\mathcal{X}>0$ and of any parity) we get the related formula:

(9)
$$\int_{\frac{\mathcal{X}+\frac{1}{2}}{N}}^{\frac{k-\frac{1}{2}}{N}} f(z)dz = \sum_{h=\mathcal{X}+1, \mathcal{X}+2, \ldots, k-1} \left[f\left(\frac{h+1}{N}\right) + 22f\left(\frac{h}{N}\right) + f\left(\frac{h-1}{N}\right)\right] \frac{1}{24N} + \varphi_2 =$$

$$= \left[f\left(\frac{\mathcal{X}+1}{N}\right) + f\left(\frac{\mathcal{X}+2}{N}\right) + \ldots + f\left(\frac{k-1}{N}\right)\right] \frac{1}{N} +$$

$$+ \left[f\left(\frac{\mathcal{X}}{N}\right) - f\left(\frac{\mathcal{X}+1}{N}\right) - f\left(\frac{k-1}{N}\right) + f\left(\frac{k}{N}\right)\right] \frac{1}{24N} + \varphi_2 ,$$

$$|\varphi_2| \leq \frac{Mf''''}{180} \frac{k-\mathcal{X}-1}{N^5} \leq \frac{Mf''''}{180} \frac{1}{N^4} .$$

10.3 We evaluate first by Simpson's formula, i.e., by (8), and in order to simplify matters we put $\mathcal{X} = 0$, $k = N$. Hence N must be even in this case

PROBLEM 10.

The function $f(z)$, z in 0, 1, is given to the extent that the values $f(h/N)$, $h = 0, 1, \ldots, N$, are stored at $N + 1$ consecutive memory locations $p, p+1, \ldots, p+N$. It is desired to evaluate the integral $\int_0^1 f(z)dz$ by means of the formula (8). ----

We could use either form of (8)'s right side for this evaluation. Using the first form leads to an induction over the odd integers $h = 1, 3, \ldots, N-1$, but requires at each step the preliminary calculation of the expression $[(f(h+1)/N) + 4f(h/N) + f((h-1)/N)]/3N$. Using the second form leads to an induction over the integers $h = 0, 1, \ldots, N$, and requires at each step the simpler expression $(2/3N)[f(h/N)]$ multiplied by a factor $\varepsilon_h = \frac{1}{2}$ or 1 or 2. This factor is best handled by variable remote connections. A detailed comparison shows that the first method is about 20% more efficient, both regarding the space required by the coded sequence and the time consumed by the actual calculation. We will nevertheless code this Problem according to the second method, because this offers a good opportunity to exemplify the use of variable remote connections.

Thus the expression to be computed is

$$J = \sum_{h=0}^{N} \varepsilon_h \frac{2}{3N} f\left(\frac{h}{N}\right) ,$$

$$\varepsilon_h \begin{cases} = \frac{1}{2} & \text{for } h = 0, N \\ = 1 & \text{for } h \neq 0, N \text{ and } h \text{ even} \\ = 2 & \text{for } h \neq 0, N \text{ and } h \text{ odd} \end{cases} .$$

This amounts to the inductive definition

$$J_{-1} = 0,$$

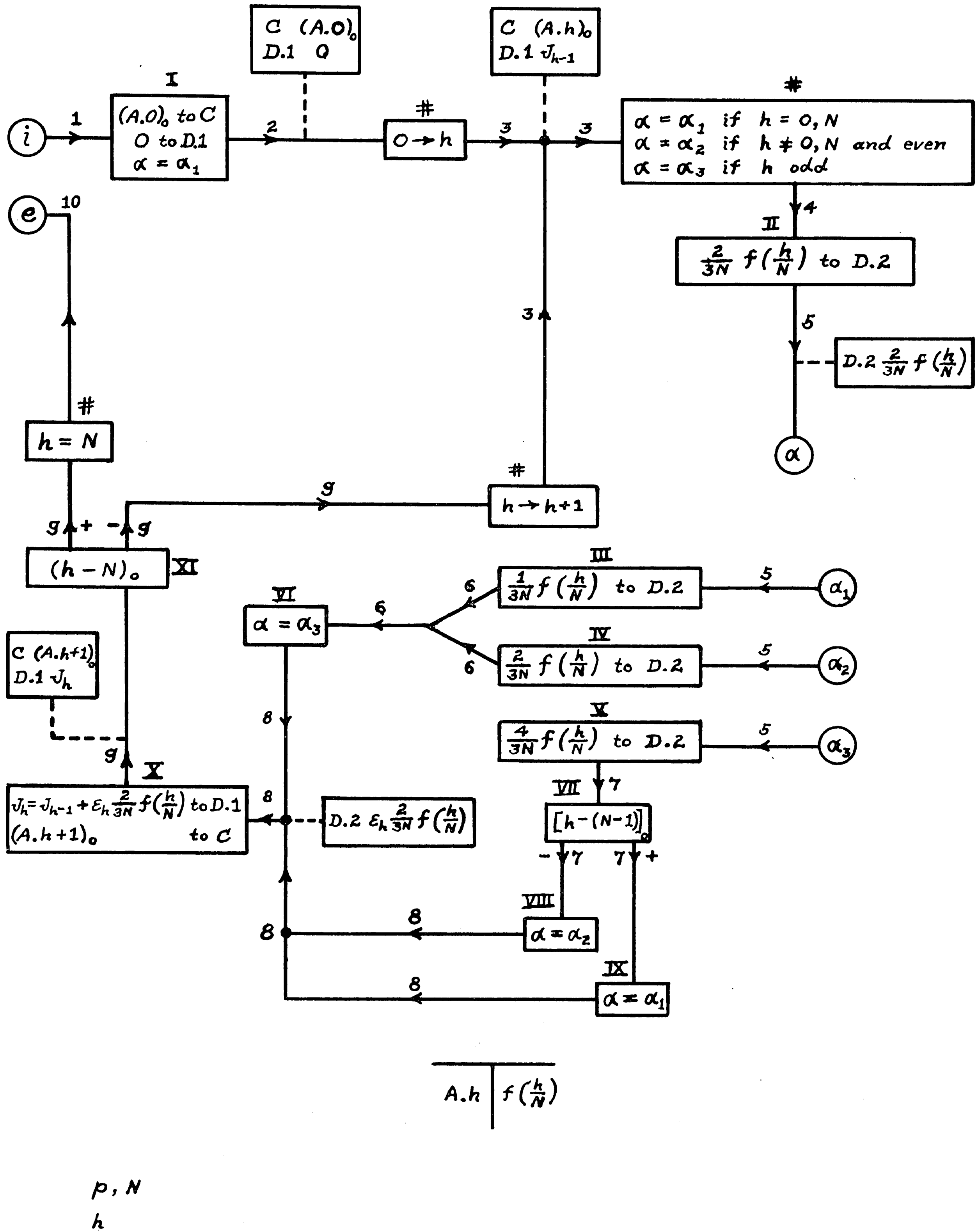
$$J_h = J_{h-1} + \varepsilon_h \frac{2}{3N} f(\frac{h}{N}) \quad \text{for } h = 0, 1, \ldots, N,$$

$$J = J_N$$

Let A be the storage area corresponding to the interval of locations from p to p + N. In this way its positions will be A.0, 1,...,N, where A.h corresponds to p + h, and stores f(h/N). These positions, however, are supposed to be parts of some other routine, already in the machine, and therefore they need not appear when we pass (at the end of the whole procedure) to the final enumeration of the coded sequence. (Cf. the analogous situation in Problem 3.)

Let B be the storage area which holds the given data (the constants) of the problem p, N. (It will actually be convenient to store p, N as p_o, $(p+N-1)_o$). Storage will also have to be provided for various other quantities (0, 1_o; it is convenient to store 2/3N; the exit-locations of the variable remote connections), these too will be accomodated in B. Next, the induction index h must be stored. h is really relevant in the combination p + h, which is a position mark (for A.h, storing f(h/N)), so we will store $(p+h)_o$ instead. (Cf. the analogous situation in Problem 3.) This will be stored at C. Finally the quantities which are processed during each inductive step will be stored in the storage area D.

These things being understood, we can now draw the flow diagram, as shown in Figure 10.1. It will prove to be convenient to store (2/3N) f(h/N) after II as well as ε_h (2/3N) f(h/N) after III and IV (but not after V) not at D.2, but in the accumulator, and to transfer it to D.2 in VI only (but also in V). The disposal of α_1, α_2, α_3, shown in IX, VIII, VI will be delayed until X, and they will be held over in the accumulator. Finally, the transfer of $(A.h+1)_o$ into C is better delayed from X until after the Cc order in XI, whereby it takes place only on the - branch issuing from XI, but this is the only branch on which it is needed.



C (A.0)₀
D.1 0
C (A.h)₀
D.1 J_{h-1}
I
(A.0)₀ to C
0 to D.1
α = α₁
#
0 → h
α = α₁ if h = 0, N
α = α₂ if h ≠ 0, N and even
α = α₃ if h odd
II
2/3N f(h/N) to D.2
D.2 2/3N f(h/N)
h = N
h → h+1
(h − N)₀
XI
III
1/3N f(h/N) to D.2
IV
2/3N f(h/N) to D.2
V
4/3N f(h/N) to D.2
VI
α = α₃
C (A.h+1)₀
D.1 J_h
X
J_h = J_{h-1} + ε_h 2/3N f(h/N) to D.1
(A.h+1)₀ to C
D.2 ε_h 2/3N f(h/N)
VII
[h − (N−1)]₀
VIII
α = α₂
IX
α = α₁
A.h
f(h/N)
p, N
h

Figure 10.1

The static coding of the boxes I-XI follows:

B.1	0					
I,1	B.1			Ac	0	
2	D.1	S		D.1	0	
B.2	p_o					
I,3	B.2			Ac	p_o	
4	C	S		C	p_o	
B.3	$(\alpha_1)_o$					
I,5	B.3			Ac	$(\alpha_1)_o$	
6	II,5	Sp		II,5	α_1	C
(to II,1)						
II,1	C			Ac	$(p+h)_o$	
2	II,3	Sp		II,3	$p+h$	R
3	-	R				
[	$p+h$	R]		R	$f(\frac{h}{N})$	
B.4	$\frac{2}{3N}$					
II,4	B.4	x		Ac	$\frac{2}{3N}\, f(\frac{h}{N})$	
5	-	C				
[	α	C]				
III,1		R		Ac	$\varepsilon_h\, \frac{2}{3N}\, f(\frac{h}{N}) = \frac{1}{3N}\, f(\frac{h}{N})$	
(to VI,1)						
IV	-					
(to VI,1)						
V,1		L		Ac	$\varepsilon_h\, \frac{2}{3N}\, f(\frac{h}{N}) = \frac{4}{3N}\, f(\frac{h}{N})$	
2	D.2	S		D.2	$\varepsilon_h\, \frac{2}{3N}\, f(\frac{h}{N})$	
(to VII,1)						
VI,1	D.2	S		D.2	$\varepsilon_h\, \frac{2}{3N}\, f(\frac{h}{N})$	
B.5	$(\alpha_3)_o$					
VI,2	B.5			Ac	$(\alpha_3)_o$	
(to X,1)						
VII,1	C			Ac	$(p+h)_o$	
B.6	$(p+N-1)_o$					
VII,2	B.6	h-		Ac	$(h-(N-1))_o$	
3	IX,1	Cc				
(to VIII,1)						
B.7	$(\alpha_2)_o$					
VIII,1	B.7			Ac	$(\alpha_2)_o$	
(to X,1)						
IX,1	B.3			Ac	$(\alpha_1)_o$	
(to X,1)						

X,1	II,5	Sp	II,5	α	C
2	D.1		Ac	J_{h-1}	
3	D.2	h	Ac	$J_h = J_{h-1} + \varepsilon_h \frac{2}{3N} f(\frac{h}{N})$	
4	D.1	S	D.1	J_h	
(to XI,1)					
XI,1	C		Ac	$(p+h)_0$	
2	B.6	h-	Ac	$(h-(N-1))_0$	
B.8	1_0				
XI,3	B.8	h-	Ac	$(h-N)_0$	
4	e	Cc			
5	C		Ac	$(p+h)_0$	
6	B.8	h	Ac	$(p+h+1)_0$	
7	C	S	C	$(p+h+1)_0$	
(to II,1)					

Note, that box IV required no coding, hence its immediate successor (VI,1) must follow directly upon its immediate predecessor. However, this box has actually no immediate predecessor; IV,1 corresponds to α_2. Hence it must be replaced there by VI,1.

The ordering of the boxes is I, II; III, VI, X, XI; V, VII, VIII; IX (IV omitted, cf. above) and X, X, II must also be the immediate successors of VIII, IX, XI, respectively. This necessitates the extra orders

VIII,2	X,1	C
IX,2	X,1	C
XI,8	II,1	C

α_1, α_2, α_3, correspond to III,1, VI,1 (instead of IV,1), V,1. In the final enumeration the three α's must obtain numbers of the same parity. This may necessitate the insertion of dummy (ineffective, irrelevant) orders in appropriate places, which we will mark *.

We must now assign B.1-8, C, D.1-2 their actual values, pair the 35 orders I,1-6, II,1-5, III,1, V,1-2, VI,1-2, VII,1-3, VIII,1-2, IX,1-2, X,1-4, XI,1-8 to 18 words, and then assign I,1-XI,8 their actual values. These are expressed in this table:

I,1-6	0 - 2'	X,1-4	7' - 9	VIII,1-2	16 - 16'
II,1-5	3 - 5	XI,1-8	9' - 13	IX,1-2	17 - 17'
III,1,*	5' - 6	V,1-2	13' - 14	B.1-8	18 - 25
VI,1-2	6' - 7	VII,1-3	14' - 15'	C	26
				D.1-2	27 - 28

Now we obtain this coded sequence:

0	18 ,	27 S	10	23 h-,	25 h-	20	5_0
1	19 ,	26 S	11	e Cc,	26	21	2/3N
2	20 ,	5 Sp	12	25 h ,	26 S	22	13_0
3	26 ,	4 Sp	13	3 C ,	L	23	$(p+N-1)_0$
4	- R ,	21 x	14	28 S ,	26	24	6_0
5	- C',	R	15	23 h-,	17 Cc	25	1_0
6	-- ,	28 S	16	24 ,	7 C'	26	--
7	22 ,	5 Sp	17	20 ,	7 C'	27	--
8	27 ,	28 h	18	0		28	--
9	27 S ,	26	19	p_0			

The durations may be estimated as follows:

I: 225 μ, II: 270 μ, III: 30 μ, V: 55 μ, VI: 75 μ: VII: 125 μ, VIII: 75 μ, IX: 75 μ, X: 150 μ, XI: 300 μ.

Total: $\mathrm{I} + \mathrm{II} \times (N+1) + \mathrm{III} \times 2 + \mathrm{VI} \times (\frac{N}{2}+1) + (\mathrm{V} + \mathrm{VII}) \times \frac{N}{2} +$

$+ \mathrm{VIII} \times (\frac{N}{2} - 1) + \mathrm{IX} + (\mathrm{X} + \mathrm{XI}) \times (N+1) =$

$= (225 + 270\,(N+1) + 60 + 75\,(\frac{N}{2}+1) + 180\,\frac{N}{2} + 75\,(\frac{N}{2} - 1) + 75 + 450\,(N+1))\ \mu =$

$= (885\,N + 1080)\ \mu \approx (.9\,N + 1.1)\ m.$

10.4 We evaluate next by the formula (9), and this time we keep ℓ, k general. We had ℓ, k = 0, 1,...,N and $k-\ell > 0$. This excludes k = 0 as well as $\ell = N$. It is somewhat more convenient to write $\ell-1$ for ℓ. Then ℓ, k = ℓ,...,N and $k-\ell \geqq 0$. Thus (9) becomes

$$(10)\qquad \int_{\frac{\ell-\frac{1}{2}}{N}}^{\frac{k-\frac{1}{2}}{N}} f(z)dz = [f(\tfrac{\ell}{N}) + f(\tfrac{\ell+1}{N}) + \ldots + f(\tfrac{k-1}{N})]\,\frac{1}{N} + [f(\tfrac{\ell-1}{N}) - f(\tfrac{\ell}{N}) - f(\tfrac{k-1}{N}) + f(\tfrac{k}{N})]\,\frac{1}{24N} + \varphi_2\,,$$

$$|\varphi_2| \leqq \frac{Mf''''}{180}\,\frac{k-\ell}{N^5} \leqq \frac{Mf''''}{180}\,\frac{1}{N^4}\,.$$

Finally the requirement $k-\ell \geqq 0$ may be dropped, since for $k-\ell \leqq 0$

$$\int_{\frac{\ell-\frac{1}{2}}{N}}^{\frac{k-\frac{1}{2}}{N}} = -\int_{\frac{k-\frac{1}{2}}{N}}^{\frac{\ell-\frac{1}{2}}{N}}\,.$$

Thus N and χ, k = 1,...,N are subject to no further restrictions.

We state accordingly:

PROBLEM 11.

Same as Problem 10, with this change. It is desired to evaluate the integral $\int_{\frac{\chi-\frac{1}{2}}{N}}^{\frac{k-\frac{1}{2}}{N}} f(z)dz$ (χ, k = 1,...,N) by means of the formula (10) (for $k \geqq \chi$, for $k < \chi$ interchange χ, k). ----

The expression to be computed is

$$J = \pm J' \quad \text{for} \begin{cases} k \geqq \chi \\ k < \chi \end{cases},$$

where J' is defined as follows: Put

$$k' = \begin{cases} k \\ \chi \end{cases}, \quad \chi' = \begin{cases} \chi \\ k \end{cases} \quad \text{for} \begin{cases} k \geqq \chi \\ k < \chi \end{cases},$$

then

$$J' = J'' + \frac{1}{24N} \left[f\left(\frac{\chi'-1}{N}\right) - f\left(\frac{\chi'}{N}\right) - f\left(\frac{k'-1}{N}\right) + f\left(\frac{k'}{N}\right)\right],$$

and

$$J'' = \frac{1}{N} \sum_{h=\chi'}^{k'-1} f\left(\frac{h}{N}\right)$$

The last equation amounts to the inductive definition

$$J_{\chi'-1} = 0,$$

$$J_h = J_{h-1} + \frac{1}{N} f\left(\frac{h}{N}\right),$$

$$J'' = J_{k'-1}.$$

The storage areas A, B, C, D and the induction index h will be treated the same way as in Problem 10, but h runs now over χ', $\chi'+1, \ldots,$ $k'-1$.

We can now draw the flow diagram as shown in Figure 10.2. It will be found convenient to store the contents of E.1-2, $(p+k')_0$, $(p+\chi')_0$, in the same place where $(p+k)_0$, $(p+\chi)_0$ are originally stored, which proves to be B.1-2. This simplifies II and reduces the memory requirements, but since we wish to have B.1-2 at the end of the routine in the same state in which they were at the beginning, it requires restoring B.1-2 in IX. VIII, on the other hand, turns out to be entirely unnecessary.

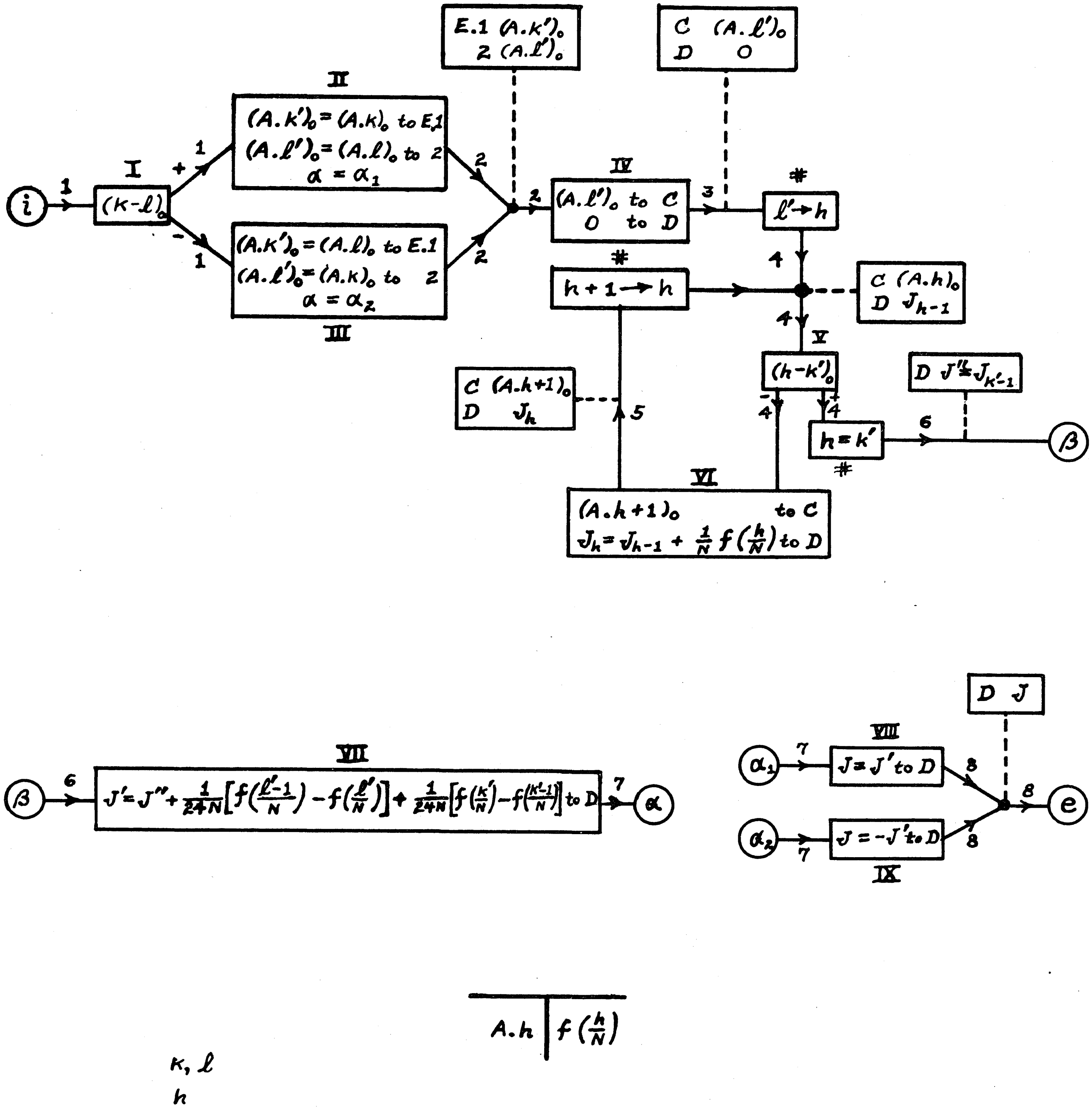

E.1 (A.k')₀
2 (A.ℓ')₀
C (A.ℓ')₀
D O
II
(A.k')₀ = (A.k)₀ to E.1
(A.ℓ')₀ = (A.ℓ)₀ to 2
α = α₁
I
(k-ℓ)₀
(A.k')₀ = (A.ℓ)₀ to E.1
(A.ℓ')₀ = (A.k)₀ to 2
α = α₂
III
IV
(A.ℓ')₀ to C
O to D
ℓ' → h
h + 1 → h
C (A.h)₀
D J_{h-1}
V
(h-k')₀
D J'' = J_{k'-1}
C (A.h+1)₀
D J_h
h = k'
VI
(A.h+1)₀ to C
J_h = J_{h-1} + (1/N) f(h/N) to D
VII
J' = J'' + (1/24N)[f((ℓ'-1)/N) - f(ℓ'/N)] + (1/24N)[f(k'/N) - f((k'-1)/N)] to D
D J
VIII
J = J' to D
J = -J' to D
IX
A.h | f(h/N)
k, ℓ
h

FIGURE 10.2

The static coding of the boxes I-IX follows:

B.1	$(p+k)_0$				
2	$(p+\ell)_0$				
I,1	B.1		Ac	$(p+k)_0$	
2	B.2	h-	Ac	$(k-\ell)_0$	
3	II,1	Cc			
(to III,1)					
B.3	$(\alpha_1)_0$				
II,1	B.3		Ac	$(\alpha_1)_0$	
2	VII,23	Sp	VII,23	α_1	C
(to IV,1)					
III,1	B.1		Ac	$(p+k)_0$	
2	s.1	S	s.1	$(p+k)_0$	
3	B.2		Ac	$(p+\ell)_0$	
4	B.1	S	B.1	$(p+\ell)_0$	
5	s.1		Ac	$(p+k)_0$	
6	B.2	S	B.2	$(p+k)_0$	
B.4	$(\alpha_2)_0$				
III,7	B.4		Ac	$(\alpha_2)_0$	
8	VII,23	Sp	VII,23	α_2	C
(to IV,1)					
IV,1	B.2		Ac	$(p+\ell')_0$	
2	C	S	C	$(p+\ell')_0$	
B.5	0				
IV,3	B.5		Ac	0	
4	D	S	D	0	
(to V,1)					
V,1	C		Ac	$(p+h)_0$	
2	B.1	h-	Ac	$(h-k')_0$	
3	VII,1	Cc			
(to VI,1)					
VI,1	C		Ac	$(p+h)_0$	
2	VI,3	Sp	VI,3	$p+h$	R
3	-	R			
[	$p+h$	R]	R	$f(\frac{h}{N})$	
B.6	$\frac{1}{N}$				
VI,4	B.6	x	Ac	$\frac{1}{N} f(\frac{h}{N})$	
5	D	h	Ac	$J_h = J_{h-1} + \frac{1}{N} f(\frac{h}{N})$	
6	D	S	D	J_h	
7	C		Ac	$(p+h)_0$	
B.7	1_0				

VI,8	B.7	h	Ac	$(p+h+1)_o$	
9	C	S	C	$(p+h+1)_o$	
(to V,1)					
VII,1	B.2		Ac	$(p+\gamma')_o$	
2	VII,6	Sp	VII,6	$p+\gamma'$	h-
3	B.7	h-	Ac	$(p+\gamma'-1)_o$	
4	VII,5	Sp	VII,5	$p+\gamma'-1$	
5	--				
[	$p+\gamma'-1$	]	Ac	$f(\frac{\gamma'-1}{N})$	
6	--	h-			
[	$p+\gamma'$	h-]	Ac	$f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N})$	
7	s.1	S	s.1	$f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N})$	
8	s.1	R	R	$f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N})$	
B.8	$\frac{1}{24N}$				
VII,9	B.8	x	Ac	$\frac{1}{24N}\ (f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N}))$	
10	D	h	Ac	$J'' + \frac{1}{24N}\ (f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N}))$	
11	D	S	D	$J'' + \frac{1}{24N}\ (f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N}))$	
12	B.1		Ac	$(p+k')_o$	
13	VII,17	Sp	VII,17	$p+k'$	h
14	B.7	h-	Ac	$(p+k'-1)_o$	
15	VII,16	Sp	VII,16	$p+k'-1$	-
16	--	-			
[	$p+k'-1$	-]	Ac	$-f(\frac{k'-1}{N})$	
17	--	h			
[	$p+k'$	h]	Ac	$f(\frac{k'}{N}) - f(\frac{k'-1}{N})$	
18	s.1	S	s.1	$f(\frac{k'}{N}) - f(\frac{k'-1}{N})$	
19	s.1	R	R	$f(\frac{k'}{N}) - f(\frac{k'-1}{N})$	
20	B.8	x	Ac	$\frac{1}{24N}\ (f(\frac{k'}{N}) - f(\frac{k'-1}{N})$	
21	D	h	Ac	$J' = J'' + \frac{1}{24N}\ (f(\frac{\gamma'-1}{N}) - f(\frac{\gamma'}{N})) + + \frac{1}{24N}\ (f(\frac{k'}{N}) - f(\frac{k'-1}{N}))$	
22	D	S	D	J'	
23	--	C			
[	α	C]			

VIII	--			
(to e)				
IX,1	D	-	Ac	J = -J'
2	D	S	D	J
3	B.1		Ac	$(p+\ell)_0 = (p+k')_0$
4	s.1	S	s.1	$(p+\ell)_0$
5	B.2		Ac	$(p+k)_0 = (p+\ell')_0$
6	B.1	S	B.1	$(p+k)_0$
7	s.1		Ac	$(p+\ell)_0$
8	B.2	S	B.2	$(p+\ell)_0$
(to e)				

Note, that the box VIII required no coding, hence its immediate successor (e) must follow directly upon its immediate predecessor. However, this box has actually no immediate predecessor; VIII,1 corresponds to α_1, which may appear (by substitution at II,2) in the C order VII,23. Hence VIII,1 must be replaced by e in α_1.

The ordering of the boxes is I, III, IV, V, VI; II; VII; IX (VIII omitted, cf. above), and IV, V, e must also be the immediate successors of II, VI, IX, respectively. This necessitates the extra orders:

II,3	IV,1	C
VI,10	V,1	C
IX,9	e	C

α_1 corresponds to VIII,1, i.e., to e (cf. above), α_2 corresponds to IX,1. This implies, as in the corresponding situation in Problem 10, that IX,1 and e must have in the final enumeration numbers of the same parity. β need not be considered, since it represents a fixed remote connection and therefore does not appear in the above detailed coding.

We must now assign B.1-8, C, D, s.1 their actual values, pair the 63 orders I,1-3, II,1-3, III,1-8, IV,1-4, V,1-3, VI,1-10, VII,1-23, IX,1-9 to 32 words, and then assign I,1-IX,9 their actual values. These are expressed in this table:

I,1-3	0 - 1	VI,1-10	9 - 13'	B.1-8	32 - 39
III,1-8	1'- 5	II,1-3	14 - 15	C	40
IV,1-4	5'- 7	VII,1-23	15'- 26'	D	41
V,1-3	7'- 8'	IX,1-9	27 - 31	s.1	42

e is supposed to have the parity of IX,1, i.e., to be unprimed.

Now we obtain this coded sequence:

0	32 ,	33 h-	14	34 ,	26 Sp'	28	32 ,	42 S
1	14 Cc,	32	15	5 C',	33	29	33 ,	32 S
2	42 S ,	33	16	18 Sp,	38 h-	30	42 ,	33 S
3	32 S ,	42	17	17 Sp',	-	31	e C ,	--
4	33 S ,	35	18	- h-,	42 S	32	$(p+k)_o$	
5	26 Sp',	33	19	42 R ,	39 x	33	$(p+\ell)_o$	
6	40 S ,	36	20	41 h ,	41 S	34	e_o	
7	41 S ,	40	21	32 ,	23 Sp'	35	27_o	
8	32 h-,	15 Cc'	22	38 h-,	23 Sp	36	0	
9	40 ,	10 Sp	23	- - ,	- h	37	$\frac{1}{N}$	
10	- R,	37 x	24	42 S ,	42 R	38	1_o	
11	41 h ,	41 S	25	39 x ,	41 h	39	$\frac{1}{24N}$	
12	40 ,	38 h	26	41 S ,	- C	40	--	
13	40 S ,	7 C'	27	41 - ,	41 S	41	--	
						42	--	

The durations may be estimated as follows:

I: 125 μ, II: 125 μ, III: 300 μ, IV: 150 μ, V: 125 μ, VI: 445 μ, VII: 1015 μ, IX: 350 μ.

Total: $I + (II \text{ or } III) + IV + V \times (|k-\ell| + 1) + VI \times |k-\ell| + VII + (VIII \text{ or } IX)$

$\text{maximum} = (125 + 300 + 150 + 125\,(|k-\ell| + 1) + 445\,|k-\ell| + 1{,}015 + 350\ \mu =$

$= (570\,|k-\ell| + 2{,}065)\ \mu$

$\text{maximum} = (570\,(N-1) + 2{,}065)\ \mu = (570\,N + 1{,}495)\ \mu \approx (.6\,N + 1.5)\ m.$

10.5 We pass now to the interpolation problem.

Lagrange's interpolation formula expresses the unique polynomial $P(x)$ of degree $M-1$, which assumes M given values $p_1,\ldots,p_M$ at M given places $x_1,\ldots,x_M$, respectively:

$$(1)\qquad P(x) = P(x_1,\ p_1;\ldots;\ x_M,\ p_M\,|\,x) = \sum_{i=1}^{M} p_i\ \frac{\prod_{j=1}^{M(j\neq i)} (x - x_j)}{\prod_{j=1}^{M(j\neq i)} (x_i - x_j)}$$

There would be no difficulty in devising a (multiply inductive) routine which evaluates the right hand side of (1) directly. This, however, seems inadvisable, except for relatively small values of M. The reason is that the denominators $\prod_{j=1}^{M(j\neq i)}$ are likely to be inadmissibly small.

This point may deserve a somewhat more detailed analysis.

From our general conditions of storage and the speed of our arithmetical organs one will be inclined to conclude that the space allotted to the storage of the functions which are evaluated by interpolation should (in a given problem) be comparable to the space occupied by the interpolation routine itself. The latter amounts to about 100 words. (Problem 12 occupies together with Problem 13.a or 13.b or 13.c 99 or 101 or 106 words, respectively, cf. 10.7 or 10.8 or 10.9, respectively. Other possible variants occupy very comparable amounts of space.) One problem will frequently use interpolation on several functions. It seems therefore reasonable to expect that each of these functions will be given at $\sim \frac{1}{3} \cdot 100 \sim 2^5$ points. (This means that the N of 10.7 will be $\sim 2^5$ -- not our present M! Note also, that the storage required in this connection is N in Problem 13.a, but 2N in Problems 13.b and 13.c.) Hence we may expect that the points $x_1, \ldots, x_M$ will be at distances of the order $\sim 2^{-5}$ between neighbors.

Hence $|x_i - x_j| \sim |i-j| \cdot 2^{-5}$, and so

$$\left| \prod_{j=1}^{M(j\neq i)} (x_i - x_j) \right| \sim (i-1)!\,(M-i)!\,2^{-5(M-1)} .$$

The round-off errors of our multiplication introduce into all these products absolute errors of the order 2^{-40}. Hence the denominators $\prod_{j=1}^{M(j\neq i)} (x_i - x_j)$, and with them the corresponding terms of the sum $\sum_{i=1}^{M}$ in (1), are affected with relative errors of the order 2^{-40}: $(i-1)!\,(M-i)!\,2^{-5(M-1)} = \frac{2^{5M-45}}{(M-1)!} \binom{M-1}{i-1}$. The average affect of these relative errors is best estimated as the relative error

$$\sqrt{\frac{1}{M} \sum_{i=1}^{M} \left[\frac{2^{5M-45}}{(M-1)!} \binom{M-1}{i-1}\right]^2} = \frac{2^{5M-45}}{(M-1)!\,\sqrt{M}} \sqrt{\sum_{i-1}^{M} \binom{M-1}{i-1}^2} =$$

$$= \frac{2^{5M-45}}{(M-1)!\,\sqrt{M}} \sqrt{\binom{2M-2}{M-1}} = \frac{2^{5M-45}}{(M-1)!\,\sqrt{M}} \sqrt{\frac{(2(M-1))!}{[(M-1)!]^2}} =$$

$$= \sqrt{\frac{(2(M-1))!}{M}} \frac{1}{[(M-1)!]^2} 2^{5M-45} .$$

On the other hand, an interpolation of degree M-1, with an interval length $\sim 2^{-5}$, is likely to have a relative precision of the order $\sim C \cdot 2^{-5M}$, where C is a moderately large number. (The function that is being interpolated is assumed to be reasonably smooth.)

Consequently the optimum relative precision ε which can be obtained with this procedure, and the optimum M that goes with it, are determined by these conditions:

$$\varepsilon \sim \sqrt{\frac{(2(M-1))!}{M}} \quad \frac{1}{[(M-1)!]} \quad 2^{5M-45} \sim C \cdot 2^{-5M} \ .$$

From this

$$f_M = \sqrt{\frac{(2(M-1))!}{M}} \cdot \frac{1}{[(M-1)!]^2} \cdot 2^{10M-45} \sim C.$$

Now we have

M	3	4	5	6	7
f_M	$8 \cdot 10^{-6}$	10^{-2}	5	$1.8 \cdot 10^{3}$	$6 \cdot 10^{5}$

The plausible values of C are in the neighborhood of M = 5, while M = 6 is somewhat high and M = 4 and 7 are extremely low and high, respectively. Hence under these conditions M = 5 (biquadratic interpolation) would seem to be normally optimal, with M = 6 a somewhat less probable possibility. We have

M	5	6
ε	$1.4 \cdot 10^{-7}$	$1.7 \cdot 10^{-6}$

It follows, therefore, that we may expect to obtain by a reasonable application of this method relative precisions of the order $\sim 10^{-6} \sim 2^{-20}$. This is, however, only the relative precision as delimited by one particular source of errors: The arithmetical (round-off) errors of an interpolation. The ultimate level of precision of the entire problem, in which this interpolation occurs, is therefore likely to be a good deal less favorable.

These things being understood, it seems likely that the resulting level of precision will be acceptable in many classes of problems, especially among those which originate in physics. There are, on the other hand, numerous and important problems where this is not desirable or acceptable, especially since it represents the loss of half the intrinsic precision of the 40 (binary) digit machine. It is therefore worthwhile to look for alternative procedures.

The obvious method to avoid the loss of (relative) precision caused in the formula (1) by the smallness of a denominator $\prod_{j=1}^{M(j\neq i)} (x_i - x_j)$, is to divide by its factors $x_i - x_j$ ($j = 1, \ldots, M$ and $j \neq i$) singly and successively. This must, however, be combined with a similar treatment of the numerators $\prod_{j=1}^{M(j\neq i)} (x-x_j)$, since they may cause a comparable loss of (relative) precision by the same mechanism. A possible alternative to (1), which eliminates both sources of error in the sense indicated, is

$$(2) \qquad P(x) = \sum_{i=1}^{M} p_i \prod_{j=1}^{M(j\neq i)} \frac{x-x_j}{x_i-x_j} .$$

(2) involves considerably more divisions than (1), but this need not be the dominant consideration. There exists, however, a third procedure, which has all the advantages of (2), and is somewhat more easily handled. Besides, its storage and induction problems are more instructive than those of (1) or (2), and for these reasons we propose to use this third procedure as the basis of our discussion.

This procedure is based on A. C. Aitken's identity

$$(3) \qquad P(x_1, p_1; \ldots; x_M, p_M \mid x) = \frac{x-x_1}{x_M-x_1} P(x_2, p_2; \ldots; x_M, p_M \mid x) + \frac{x_M-x}{x_M-x_1} P(x_1, p_1; \ldots; x_{M-1}, p_{M-1} \mid x) .$$

Since (3) replaces an M-point interpolation by two M-1 point interpolations, it is clearly a possible basis for an inductive procedure. It might seem, however, that the reduction from an M-point interpolation to one-point ones (i.e. to constants) will involve 2^{M-1} steps, which would be excessive, since (2) is clearly an M(M-1) step process. However, (3) removes either extremity (x_1 or x_M) of the point system $x_1,\ldots,x_M$; hence iterating it can only lead to point systems $x_i,\ldots,x_j$ ($i, j = 1,\ldots,M$, $i \leq j$, we will write $j = i+h-1$), of which there are only $\frac{M(M+1)}{2}$; i.e., $\frac{M(M-1)}{2}$, not counting the one-point systems. Hence (3) is likely to lead to something like an $\frac{M(M-1)}{2}$ step process.

Regarding the sizes we assume that $x_1,\ldots,x_M$ as well as x and $p_1,\ldots,p_M$ lie in the interval -1, 1. We need, furthermore, that the differences $x_{i+h}-x_i$, $x-x_i$, $p_{i+h}-p_i$ also lie in the interval -1, 1, and it is even necessary in view of the particular algebraical routine that we use, to have all differences $p_{i+h}-p_i$ (absolutely) smaller than the corresponding $x_{i+h}-x_i$. (Cf. VII,19.) I.e., we must use appropriate size adjusting factors for the p_i to secure this "Lipschitz condition". The same must be postulated for the differences $P^h_{i+1}(x)-P^h_i(x)$ of the intermediate interpolation polynomials. All of this might be circumvented to various extents in various ways, but this would lead us deeply into the problems of polynomial interpolation which are not our concern here.

We now state:

PROBLEM 12.

The variable values $x_1, \ldots, x_M$ ($x_1 < x_2 < \ldots < x_m$) and the function values $p_1, \ldots, p_M$ are stored at two systems of M consecutive memory locations each: q, q+1, ..., q+M-1 and p, p+1, ..., p+M-1. The constants of the problem are p, q, M, x, and they are stored at four given memory locations. It is desired to interpolate this function for the value x of the variable, using Lagrange's formula. The process of reduction based on the identity (3) is to be used. ----

It is clear from the previous discussion, that we have to form the family of interpolants

$$(4) \qquad P_i^h(x) = P(x_i, p_i; \ldots; x_{i+h-1}, p_{i+h-1} \mid x) \qquad \text{for } i = 1, \ldots, M, \quad h = 1, \ldots, M-i+1 .$$

Applying (3) to $P(x_i, p_i; \ldots; x_{i+h}, p_{i+h} \mid x)$ instead of $P(x_1, p_1; \ldots; x_M, p_M \mid x)$ gives

$$P_i^{h+1}(x) = \frac{x \quad -x_i}{x_{i+h}-x_i} \quad P_{i+1}^h(x) + \frac{x_{i+h}-x}{x_{i+h}-x_i} \quad P_i^h(x) \quad ,$$

or equivalently

$$(5) \qquad P_i^{h+1}(x) = P_i^h(x) + \frac{x \quad -x_i}{x_{i+h}-x_i} \quad (P_{i+1}^h(x) - P_i^h(x)) \quad .$$

Combining this with

$$(6) \qquad P_i^1(x) = p_i$$

and

$$(7) \qquad P(x) = P(x_1, p_1; \ldots; x_M, p_M \mid x) = P_1^M(x) \quad ,$$

we have a (doubly) inductive scheme to calculate P(x).

. Let A and B be the storage areas corresponding to the two intervals of locations from p to p+M-1 and from q to q+M-1. In this way their positions will be A.1,...,M and B.1,...,M, where A.i and B.i correspond to p+i-1 and q+i-1, and store p_i and x_i, respectively. As in the Problems 3, 10, and 11, the positions of A and B need not be shown in the final enumeration of the coded sequence.

The primary induction will clearly begin with the $p_1, \ldots, p_M$, i.e., $P_1^1(x), \ldots, P_M^1(x)$, stored at A, and obtain from these $P_1^2(x), \ldots, P_{M-1}^2(x)$; then from these the $P_1^3(x), \ldots, P_{M-2}^3(x)$; from these the $P_1^4(x), \ldots, P_{M-3}^4(x)$; etc., etc., to conclude with $P_1^M(x)$, i.e., with the desired P(x). These successive stages correspond to h = 1, 2, 3,...,M, respectively. This h is the primary induction index.

In passing from h to h+1, i.e., from $P_1^h(x), \ldots, P_{M-h+1}^h(x)$ to $P_1^{h+1}(x), \ldots, P_{M-h}^{h+1}(x)$, the $P_i^{h+1}(x)$ have to be formed successively for $i = 1, \ldots, M-h$. This i is the secondary induction index.

There is clearly need for a storage area C which holds $P_1^h(x), \ldots, P_{M-h+1}^h(x)$ at each stage h. As the transition to stage h+1 takes place, each $P_i^h(x)$ will be replaced by $P_i^{h+1}(x)$, successively for all $i = 1, \ldots, M-h$. Hence the capacity required for C is M-h+1 during the stage h, but for h = 1 the $P_i^1(x) = p_i$ are still in A, and need not appear in C. Hence the maximum capacity for C is M-1. Accordingly, let C correspond to the interval of locations from r to r+M-2. In this way its positions will be C.1,...,M-1, where C.i corresponds to r+i-1, and stores $P_i^h(x)$. The positions of the area C need not be shown in the final enumeration of the coded sequence. All that is necessary is that they be available (i.e. empty or irrelevantly occupied) when the instructions of the coded sequence are being carried out by the machine.

As stated above, the $P_i^h(x)$ occupying C.i have necessarily $h > 1$. To give more detail: $P_i^h(x)$ moves into the location C.i at the end of the i-th step of stage h-1, and remains there during the remainder (the M-h+1-i last steps) of stage h-1 and during the i first steps of stage h.

Further storage capacities are required as follows: The given data (the constants) of the problem, p, q, r, M, x, will be stored in the storage area D. (It will be convenient to store them as p_0, $(q-p-1)_0$, r_0, $(M-1)_0$, x.) Storage will also have to be provided for various other fixed quantities (1_0, the exit-locations of the variable remote connections), these too will be accommodated in D. Next, the two induction indices h, i, will have to be stored. As before, they are both relevant as position marks, and it will prove convenient to store in their place $(M-h)_0$, $(p+i)_0$ (i.e., $(A.i+1)_0$). These will be stored in the storage area E. Finally, the quantities which are processed during each inductive step will be stored in the storage area F.

We can now draw the flow diagram, as shown in Figure 10.3. The variable remote connections α and β are necessary, in order to make the differing treatments required for h = 1 and for h > 1 possible: In the first case the $P_i^h(x)$ are equal to p_i and come from A, in the second case they come from C (cf. above). It should be noted that in this situation our flow diagram rules impose a rather detailed showing of the contents of the storage area C.

The actual coding contains a number of minor deviations from the flow diagram, inasmuch as it is convenient to move a few operations from the box in which they are shown to an earlier box. Since we had instances of this already in earlier problems, we will not discuss it here in detail. On the other hand, some further condensations of the coding, which are possible, but deviate still further from the flow diagram, will not be considered here.

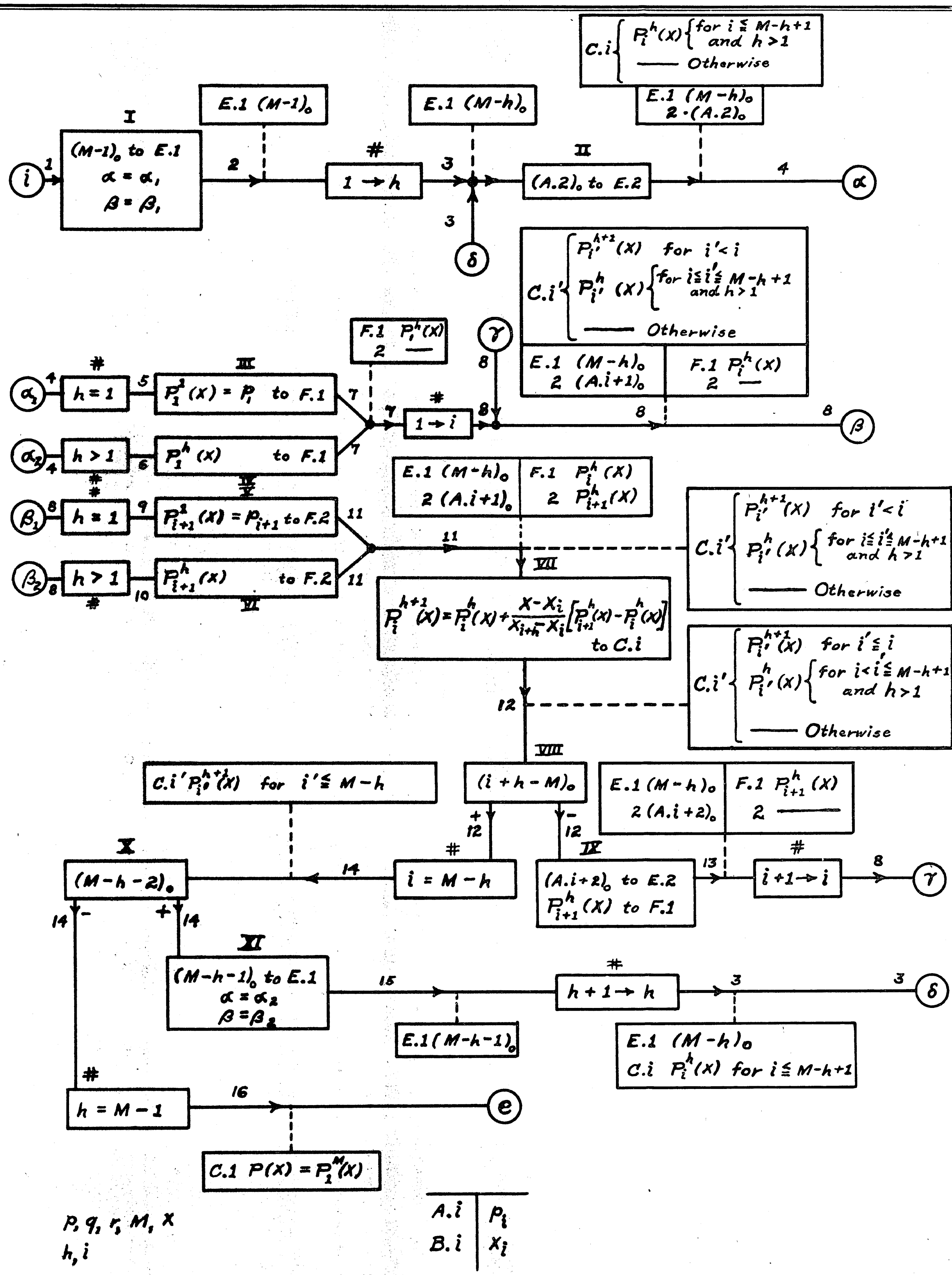

C.i $P_i^h(X)$ for $i \leq M-h+1$ and $h>1$ —— Otherwise
E.1 $(M-1)_0$
E.1 $(M-h)_0$
E.1 $(M-h)_0$ 2 $\cdot(A.2)_0$
I
$(M-1)_0$ to E.1 $\alpha = \alpha_1$ $\beta = \beta_1$
$1 \rightarrow h$
II
$(A.2)_0$ to E.2
C.i′ $P_{i'}^{h+1}(X)$ for $i'<i$ $P_{i'}^h(X)$ for $i \leq i' \leq M-h+1$ and $h>1$ —— Otherwise
E.1 $(M-h)_0$ 2 $(A.i+1)_0$
F.1 $P_i^h(X)$ 2 ——
F.1 $P_i^h(X)$ 2 ——
III
$h=1$
$P_1^1(X) = P_1$ to F.1
$h>1$
$P_1^h(X)$ to F.1
$1 \rightarrow i$
IV
V
$h=1$
$P_{i+1}^1(X) = P_{i+1}$ to F.2
$h>1$
$P_{i+1}^h(X)$ to F.2
VI
E.1 $(M-h)_0$ 2 $(A.i+1)_0$
F.1 $P_i^h(X)$ 2 $P_{i+1}^h(X)$
C.i′ $P_{i'}^{h+1}(X)$ for $i'<i$ $P_{i'}^h(X)$ for $i \leq i' \leq M-h+1$ and $h>1$ —— Otherwise
VII
$P_i^{h+1}(X) = P_i^h(X) + \frac{X-X_i}{X_{i+h}-X_i}\left[P_{i+1}^h(X) - P_i^h(X)\right]$ to C.i
C.i′ $P_{i'}^{h+1}(X)$ for $i' \leq i$ $P_{i'}^h(X)$ for $i < i' \leq M-h+1$ and $h>1$ —— Otherwise
VIII
$(i+h-M)_0$
C.i′ $P_{i'}^{h+1}(X)$ for $i' \leq M-h$
E.1 $(M-h)_0$ 2 $(A.i+2)_0$
F.1 $P_{i+1}^h(X)$ 2 ——
X
$(M-h-2)_0$
$i = M-h$
IX
$(A.i+2)_0$ to E.2 $P_{i+1}^h(X)$ to F.1
$i+1 \rightarrow i$
XI
$(M-h-1)_0$ to E.1 $\alpha = \alpha_2$ $\beta = \beta_2$
$h+1 \rightarrow h$
E.1 $(M-h-1)_0$
E.1 $(M-h)_0$ C.i $P_i^h(X)$ for $i \leq M-h+1$
$h = M-1$
C.1 $P(X) = P_1^M(X)$
P, q, r, M, X
h, i
A.i | P_i
B.i | X_i

FIGURE 10.3

The static coding of the boxes I-XI follows:

D.1	$(M-1)_0$					
I,1	D.1			Ac	$(M-1)_0$	
2	E.1	S		E.1	$(M-1)_0$	
D.2	$(\alpha_1)_0$					
I,3	D.2			Ac	$(\alpha_1)_0$	
4	II,4	Sp		II,4	α_1	C
D.3	$(\beta_1)_0$					
I,5	D.3			Ac	$(\beta_1)_0$	
6	III,6	Sp		III,6	β_1	C
(to II,1)						
D.4	p_0					
5	1_0					
II,1	D.4			Ac	p_0	
2	D.5	h		Ac	$(p+1)_0$	
3	E.2	S		E.2	$(p+1)_0$	
4	--	C				
[	α	C	]			
III,1	D.4			Ac	p_0	
2	III,3	Sp		III,3	p	
3	--					
[	p		]	Ac	$p_1^1(x) = p_1$	
4	F.1	S		F.1	p_1 if reached from III,3	
					$p_1^h(x)$ if reached from IV	
5	E.2			Ac	$(p+1)_0$ if reached from III,4	
					$(p+i)_0$ if reached from IX (i.e. γ)	
6	--	C				
[	β	C	]			
D.6	r_0					
IV,1	D.6			Ac	r_0	
2	IV,3	Sp		IV,3	r	
3	--					
[	r		]	Ac	$p_1^h(x)$	
(to III,4)						
V,1	V,2	Sp		V,2	p+i	
2	--					
[	p+i		]	Ac	p_{i+1}	
3	F.2	S		F.2	p_{i+1}	
(to VII,1)						
VI,1	D.4	h-		Ac	i_0	
2	D.6	h		Ac	$(r+i)_0$	
3	VI,4	Sp		VI,4	r+i	
4	--					
[	r+i		]	Ac	$p^h_{i+1}(x)$	
5	F.2	S		F.2	$p^h_{i+1}(x)$	

(to VII,1)

VII,1	E.2		Ac	$(p+i)_o$	
2	D.4	h-	Ac	i_o	
3	D.6	h	Ac	$(r+i)_o$	
4	D.5	h-	Ac	$(r+i-1)_o$	
5	VII,25	Sp	VII,25	$r+i-1$	S
6	E.2		Ac	$(p+i)_o$	
D.7	$(q-p-1)_o$				
VII,7	D.7	h	Ac	$(q+i-1)_o$	
8	VII,15	Sp	VII,15	$q+i-1$	h-
9	VII,21	Sp	VII,21	$q+i-1$	h-
10	D.1	h	Ac	$(q+i+M-2)_o$	
11	E.1	h-	Ac	$(q+i+h-2)_o$	
12	D.5	h	Ac	$(q+i+h-1)_o$	
13	VII,14	Sp	VII,14	$q+i+h-1$	
14	--				
[	$q+i+h-1$]		Ac	x_{i+h}	
15	--	h-			
[	$q+i-1$	h-]	Ac	$x_{i+h}-x_i$	
16	s.1	S	s.1	$x_{i+h}-x_i$	
17	F.2		Ac	$P^h_{i+1}(x)$	
18	F.1	h-	Ac	$P^h_{i+1}(x)-P^h_i(x)$	
19	s.1	+	R	$\frac{P^h_{i+1}(x)-P^h_i(x)}{x_{i+h}-x_i}$	
D.8	x				
VII,20	D.8		Ac	x	
21	--	h-			
[	$q+i-1$	h-]	Ac	$x-x_i$	
22	s.1	S	s.1	$x-x_i$	
23	s.1	x	Ac	$\frac{x-x_i}{x_{i+h}-x_i}(P^h_{i+1}(x)-P^h_i(x))$	
24	F.1	h	Ac	$P^{h+1}_i(x) = P^h_i(x) + \frac{x-x_i}{x_{i+h}-x_i}(P^h_{i+1}(x)-P^h_i(x))$	
25	--	S			
[	$r+i-1$	S]	C.i	$P^{h+i}_i(x)$	

(to VIII,1)

VIII,1	E.2		Ac	$(p+i)_0$	
2	D.5	h	Ac	$(p+i+1)_0$	
3	E.2	S	E.2	$(p+i+1)_0$	
4	E.1	h-	Ac	$(p+i+h-M+1)_0$	
5	D.5	h-	Ac	$(p+i+h-M)_0$	
6	D.4	h-	Ac	$(i+h-M)_0$	
7 (to IX,1)	X,1	Cc			
IX,1	F.2		Ac	$P^h_{i+1}(x)$	
2 (to III,5)	F.1	S	F.1	$P^h_{i+1}(x)$	
X,1	E.1		Ac	$(M-h)_0$	
2	D.5	h-	Ac	$(M-h-1)_0$	
3	E.1	S	E.1	$(M-h-1)_0$	
4	D.5	h-	Ac	$(M-h-2)_0$	
5 (to e)	XI,1	Cc			
D.9	$(\alpha_2)_0$				
XI,1	D.9		Ac	$(\alpha_2)_0$	
2	II,4	Sp	II,4	α_2	C
D.10	$(\beta_2)_0$				
XI,3	D.10		Ac	$(\beta_2)_0$	
4 (to II,1)	III,6	Sp	III,6	β_2	C

The ordering of the boxes is I, II; III; IV; V, VII, VIII, IX; X; XI; VI, and VII, e, II must also be the immediate successors of VI, X, XI, respectively, and III,4 and III,5 must be the immediate successors of IV and IX, respectively. This necessitates the extra orders

VI,6	VII,1	C
X,6	e	C
XI,5	II,1	C

and

IV,4	III,4	C
IX,3	III,5	C

α_1, α_2, β_1, β_2 correspond to III,1, IV,1, V,1, VI,1, respectively. Hence in the final enumeration III,1, IV,1 must have the same parity, and V,1, VI,1 must have the same parity.

We must now assign D.1-10, E.1-2, F.1-2, s.1 their actual values, pair the 75 orders I,1-6, II,1-4, III,1-6, IV,1-4, V,1-3, VI,1-6, VII,1-25, VIII,1-7, IX,1-3, X,1-6, XI,1-5 to 38 words, and then assign I,1-XI,5 their final values. These are expressed in this table:

I,1-6	0 -2'	VII,1-25	11'-23'	VI,1-6	35 -37'
II,1-4	3 -4'	VIII,1-7	24 -27	D.1-10	38 -47
III,1-6	5 -7'	IX,1-3	27'-28'	E.1-2	48 -49
IV,1-4	8 -9'	X,1-6	29 -31'	F.1-2	50 -51
V,1-3	10 -11	XI,1-5,*	32 -34'	s.1	52

Now we obtain this coded sequence:

0	38 , 48 S	18	- , - h-	36	36 Sp', -
1	39 , 4 Sp'	19	52 S , 51	37	51 S , 11 C'
2	40 , 7 Sp'	20	50 h-, 52 ÷	38	$(M-1)_o$
3	41 , 42 h	21	45 , - h-	39	5_o
4	49 S , - C	22	52 S , 52 x	40	10_o
5	41 , 6 Sp	23	50 h , - S	41	p_o
6	- , 50 S	24	49 , 42 h	42	1_o
7	49 , - C	25	49 S , 48 h-	43	r_o
8	43 , 9 Sp	26	42 h-, 41 h-	44	$(q-p-1)_o$
9	- , 6 C'	27	29 Cc, 51	45	x
10	10 Sp', -	28	50 S , 7 C	46	8_o
11	51 S , 49	29	48 , 42 h-	47	35_o
12	41 h-, 43 h	30	48 S , 42 h-	48	-
13	42 h-, 23 Sp'	31	32 Cc, e C	49	-
14	49 , 44 h	32	46 , 4 Sp'	50	-
15	18 Sp', 21 Sp'	33	47 , 7 Sp'	51	-
16	38 h , 48 h-	34	3 C , -	52	-
17	42 h , 18 Sp	35	41 h-, 43 h		

The durations may be estimated as follows:

I: 225 μ, II: 150 μ, III: 225 μ, IV: 275 μ, V: 125 μ, VI: 225 μ, VII: 1140 μ, VIII: 275 μ, IX: 200 μ, X: 225 μ, XI: 200 μ.

Total:

$$I + II \times (M-1) + III + IV \times (M-2) + V \times (M-1) + VI \times \frac{(M-1)(M-2)}{2} +$$

$$+ (VII + VIII) \frac{M(M-1)}{2} + IX \times \frac{(M-1)(M-2)}{2} + X \times (M-1) + XI \times (M-2) =$$

$$= (225 + 150\ (M-1) + 225 + 275\ (M-2) + 125\ (M-1) + 225 \frac{(M-1)(M-2)}{2} +$$

$$+ 1{,}415 \frac{M(M-1)}{2} + 200 \frac{(M-1)(M-2)}{2} + 225\ (M-1) + 200\ (M-2))\ \mu =$$

$$= (920\ M^2 - 370\ M - 575)\ \mu \approx (.9\ M^2 - .4\ M - .6)\ m.$$

10.6 We now pass to the problem of interpolating a tabulated function based on the coding of Lagrange's interpolation formula in Problem 12.

PROBLEM 13.

The variable values $y_1, \ldots, y_N$ ($y_1 < y_2 < \ldots < y_N$) and the function values $q_1, \ldots, q_N$ are stored at two systems of N consecutive memory locations each: $\bar{q}, \bar{q}+1, \ldots, \bar{q}+N-1$ and $\bar{p}, \bar{p}+1, \ldots, \bar{p}+N-1$. The constants of the problem are $\bar{p}$, $\bar{q}$, N, M, y, to be stored at five given memory locations. It is desired to interpolate this function for the value y of the variable, using Lagrange's formula for the M points y_i nearest to y. ----

The problem should be treated differently, according to whether the $y_1, \ldots, y_N$ are or are not to be equidistant.

PROBLEM 13.a.

$y_1, \ldots, y_N$ are equidistant, i.e., $y_i = a + \frac{i-1}{N-1}(b-a)$. In this case only $y_1 = a$ and $y_N = b$ need be stored. ----

PROBLEM 13.b.

$y_1, \ldots, y_N$, are unrestricted. ----

In both cases our purpose is to reduce Problem 13 to Problem 12, with $x_1, \ldots, x_M$ equal to $y_k, \ldots, y_{k+M-1}$, and $p_1, \ldots, p_M$ equal to $q_k, \ldots, q_{k+M-1}$, where $k = 1, \ldots, N-M+1$ is so chosen that the $y_k, \ldots, y_{k+M-1}$ lie as close to y as possible.

10.7 We consider first Problem 13.a.

In this case the definition of k, as formulated at the end of 10.6, amounts to this: The remoter one of $y_k = a + \frac{k-1}{N-1}(b-a)$ and $y_{k+M-1} = a + \frac{k+M-2}{N-1}(b-a)$ should lie as close as possible to y. This is equivalent to requiring that their mean, $\frac{1}{2}(y_k + y_{k+M-1}) = a + \frac{2k+M-3}{2(N-1)}(b-a)$, lie as close as possible to y, i.e., that k lie as close as possible to $(N-1)\frac{y-a}{b-a} - \frac{M-3}{2}$.

There are various ways to find this $k = \bar{k}$, based on iterative trial and error procedures. Since we will have to use a method of this type in connection with Problem 13.b, we prefer a different one at this occasion.

This method is based on the function $\{z\}$, which denotes the integer closest to z. Putting

$$k^* = \left\{(N-1)\frac{y-a}{b-a} - \frac{M-3}{2}\right\},$$

we have clearly

$$\bar{k} \begin{cases} = 1 & \text{for } k^* \leqq 1, \\ = k^* & \text{for } 1 \leqq k \leqq N\text{-}M\text{+}1, \\ = N\text{-}M\text{+}1 & \text{for } k^* \geqq N\text{-}M\text{+}1 \end{cases}$$

Now the multiplication order (11, Table II), permits us to obtain $\{z\}$ directly. Indeed, the round off rule (cf. the discussion of the order in question) has the effect that when a product uv is formed, the accumulator will contain $\overrightarrow{uv} = 2^{-39}\{2^{39}uv\}$, and the arithmetical register will contain $\overrightarrow{uv} = 2^{39}uv - \{2^{39}uv\}$. Hence putting $u = 2^{-39}(N-1)$ and $v = \frac{y-a}{b-a} - \frac{M-3}{2(N-1)}$ will produce $\overrightarrow{uv} = 2^{-39}\{2^{39}uv\} = 2^{-39}k^*$ in the accumulator.

After k^* and $\bar{k}$ have been obtained, we can utilize the routine of Problem 12 to complete the task. This means, that we propose to use the coded sequence 0-52 of 10.5, and that we will adjust the coded sequence that will be formed here, so that it can be used in conjunction with that one of 10.5.

Among the constants of Problem 12 only p, q, and x need be given values which correspond to the new situation. x is clearly our present y. $p,\ldots,p+M-1$ are the positions of the $p_1,\ldots,p_M$ of Problem 12, i.e., the positions of our present $q_{\bar{k}},\ldots,q_{\bar{k}+M-1}$, i.e., they are $\bar{p}+\bar{k}-1,\ldots,\bar{p}+\bar{k}+M-2$. Hence $p = \bar{p}+\bar{k}-1$. $q,\ldots,q+M-1$ are the positions of the $x_1,\ldots,x_M$ of Problem 12, i.e., the positions of our present $y_{\bar{k}},\ldots,y_{\bar{k}+M-1}$. Since we determined in the formulation of Problem 13.a, that only $y_1 = a$ and $y_N = b$ will be stored, but not the entire sequence $y_1,\ldots,y_N$, this means that the desired sub-sequence $y_{\bar{k}},\ldots,y_{\bar{k}+M-1}$ does not exist anywhere ab initio.

Consequently q may have any value, all that is needed is that the positions $q,\ldots,q+M-1$ should be available and empty (or irrelevantly occupied) when the coded sequence that we are going to formulate begins to operate. This sequence must then form

$$x_i = y_{\bar{k}+i-1} = a + \frac{\bar{k}+i-2}{N-1}(b-a)$$

and place it into the position $q+i-1$ for all $i = 1,\ldots,M$.

It might seem wasteful to form x_i first, then store it at $q+i-1$, and finally obtain it from there by transfer when it is needed, i.e. during the period VII,1-19 of the coded sequence of 10.5. One might think that it is simpler to form x_i when it is actually needed, i.e., in VII,1-19 as stated above; the quantities needed are more specifically x_i, x_{i+h}, in the combination $\frac{x - x_i}{x_{i+h} - x_i}$), and thus avoid the transfers and the storage. It is easy to see, however, that the saving thus effected is altogether negligible, essentially because the size of our problem is proportional to M^2 (cf. the end of 10.5), while the number of steps required in forming and transferring the x_i's in the first mentioned way is only proportional to M. (Remember that M is likely to be $\geqq 7$, cf. the beginning of 10.5.) It does therefore hardly seem worthwhile to undertake those changes of the coded sequence of 10.5 which the second procedure would necessitate, and we will adhere to the first procedure.

In assigning letters to the various storage areas to be used, it must be remembered that the coded sequence that we are now developing is to be used in conjunction with (i.e., as a supplement to) the coded sequence of 10.5. The latter requires storage areas of various types: D-F, which are incorporated into its final enumeration (they are 38-51 of the 0-52 of 10.5); A (i.e., $p,\ldots,p+M-1$), which will be part of our present A (i.e., of $\bar{p},\ldots,\bar{p}+N-1$, it will be $\bar{p}+\bar{k}-1,\ldots,\bar{p}+\bar{k}+M-2$; B (i.e., $q,\ldots,q+M-1$), which may be at any available place; and C (i.e., $r,\ldots,r+M-2$), which, too, may be at any available place. Therefore we can, in assigning letters to the various storage areas of the present coding, disregard those of 10.5, with the exception of B, C. Furthermore, there will be no need to refer here to C (of 10.5), since the coded sequence of 10.5 assumes the area C to be irrelevantly occupied. It will be necessary, however, to refer to B (of 10.5), since it is supposed to contain the $x_i = y_{\bar{k}+i-1}$ $(i = 1,\ldots,M)$ of the problem. We will therefore think of the letters which are meant to designate storage areas of the coded sequence of 10.5 as being primed. This is, according to the above, an immediate necessity for B. We can now assign freely unprimed letters to the storage areas of the present coding.

Let A be the storage area corresponding to the interval from $\bar{p}$ to $\bar{p}+N-1$. In this way its positions will be A.1,...,N, where A.i corresponds to $\bar{p}+i-1$ and stores q_i. As in previous problems, the positions of A need not be shown in the final enumeration of the coded sequence.

The given data of the problem are $\bar{p}$, M, N, a, b, y, also the q, r of the coded sequence of 10.5. M, r are already stored there (as $(M-1)_0$, r_0 at 38, 43); and y, too (it coincides with x at 45). q, however, occurs only in combination with $\bar{k}$ (as $(q-p-1)_0 = (q-\bar{p}-\bar{k})_0$ at 44) and p, of course, contains $\bar{k}$ (as $p_0 = (\bar{p}+\bar{k}-1)_0$ at 41); and $\bar{k}$ originates in the machine. Hence, 41, 44 must be left empty (or, rather, irrelevantly occupied) when our present coded sequence begins to operate, and they must be appropriately substituted by its operations. q, however, must be stored as one of the constants of our present problem. Hence the constants requiring storage now (apart from M, r, y which we stored in 0-52, cf. above) are $\bar{p}$, q, N, a, b. They will be stored in the storage area B. (It will be convenient to store them as a, b, $(N-1)_0$, $(\bar{p}-1)_0$, $(q-1)_0$. We will also need 2^{-39} and 1_0; we will store the former in B and get the latter from 42.) The induction index i is a position mark; it will be stored as $(q+i-1)_0$ (i.e., $(B'.i)_0$) in the storage area C. The quantities which are processed during each inductive step will also be stored in the storage area C.

Our task is to calculate $\bar{k}$; to substitute $p_0 = (p+\bar{k}-1)_0$ and $(q-p-1)_0 = (\bar{q}-\bar{p}-k)_0$ into 41 and 44; and then to transfer $x_i = y_{\bar{k}+i-1}$ from A.$\bar{k}$+i-1 to B'.i. This latter operation is inductive. Finally, the control has to be sent, not to e, but to the beginning of 0-52, i.e., to 0.

We can now draw the flow diagram, as shown in Figure 10.4.

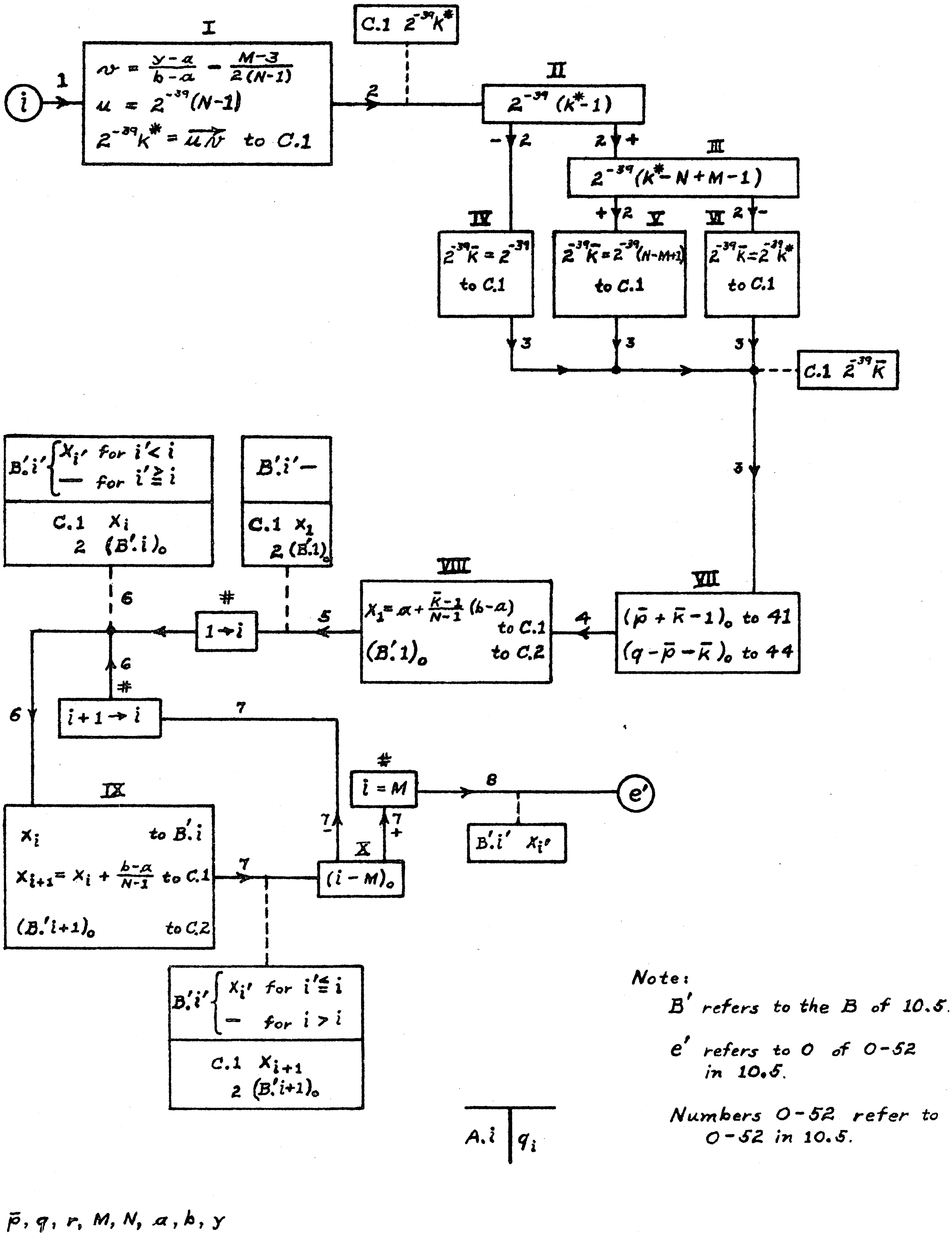

I
II
III
IV
V
VI
VII
VIII
IX
X
C.1 2^-39 k*
2^-39 (k*-1)
2^-39 (k*-N+M-1)
to C.1
C.1 2^-39 k̄
(p̄ + k̄ - 1)_0 to 41
(q - p̄ - k̄)_0 to 44
(B'.1)_0 to C.2
1 → i
i + 1 → i
i = M
(i - M)_0
B'.i' X_i'
to B'.i
(B'.i+1)_0 to C.2
e'
Note:
B' refers to the B of 10.5.
e' refers to 0 of 0-52 in 10.5.
Numbers 0-52 refer to 0-52 in 10.5.
A.i q_i
p̄, q, r, M, N, a, b, y
i

Figure 10.4

The actual coding will again deviate in some minor respects from the flow diagram, as in previous instances. In connection with this we will also need some extra storage capacity in C (C.1.1).

A matter of somewhat greater importance is this: We have noted before that since our machine recognizes numbers between -1 and 1 only, integers I must be stored in some other form. Frequently the form of a position mark, I_0, is per se more natural than any other (cf. 8.2); occasionally $2^{-39}I$ can be fitted more easily to the algebraical use to be made of I (cf. I in the present coding); sometimes $\frac{1}{I}$ is most convenient. (We could, of course, add to this list, but the three above forms seem to be the basic ones.) The transitions between these three forms are easily effected by using multiplications and divisions, but it seems natural to want to achieve the transitions between the two first ones (I_0 and $2^{-39}I$) in a more direct way.

This can be achieved by means of the partial substitution orders 18, 19 of Table II (x Sp, x Sp'), if they are modified as indicated in the Remark immediately preceding this chapter. We propose to use these orders now for arithmetical rather than logical (substitution) purposes, for positions x which contain no orders at all, but which are storing numbers in transit. Specifically: With I_0 in the accumulator, x Sp' produces $2^{-39}I$ at the position x; with $2^{-39}I$ in the accumulator x Sp produces $2^{-19}I$ at the position x, and a subsequent x h produces I_0 in the accumulator.

We mention that $\frac{1}{I}$ can be obtained from I_0 or $2^{-39}I$ by dividing them into 1_0 or 2^{-39}, respectively, and this without more than the usual loss in precision: Indeed, our division $\rho : \sigma$ is precise within an error 2^{-39}, no matter how small σ is (subject, of course, to the condition $|\sigma| > |\rho|$), provided that ρ, σ are given exactly. (If they are not given exactly, then their errors are amplified by $\frac{1}{\sigma}$, $\frac{\rho}{\sigma^2}$, respectively. In this case a small σ is dangerous, even though $|\sigma| > |\rho|$.) In the present case ρ, σ are given exactly. Forming 1_0, I_0, as well as 2^{-39}, $2^{-39}I$, involves no round-offs.

These methods will be used in our present coding: For transitions between I_0 and $2^{-39}I$, cf. I,22-23, III,3-4; VII,1-3; For forming reciprocals: $\frac{1}{N-1}$ from $(N-1)_0$ as 1_0: $(N-1)_0$ in VIII,4-5, and the very similar case of $\frac{M-3}{2(N-1)}$ from $(M-3)_0$, $(N-1)_0$ as $(M-3)_0$: $2(N-1)_0$ in I,9-15. Regarding $\frac{1}{N-1}$ we also note this: We need $\frac{b-a}{N-1}$ in VIII. We will form $\frac{1}{N-1} = 1_0$: $(N-1)_0$ first and $\frac{b-a}{N-1} = \frac{1}{N-1} \times (b-a)$ **afterwards.** Forming $1_0 \times (b-a)$ first and $\frac{b-a}{N-1} = [1_0 \times (b-a)]: (N-1)_0$ afterwards would lead to a serious loss of precision, since $1_0 \times (b-a)$ plays the role of ρ above, and as it involves a round-off it is not given exactly, and hence may cause a loss of precision as indicated there.

We will have to refer in our present coding repeatedly to 0-52 in 10.5. It should therefore be remembered that this coded sequence is now supposed to be changed insofar that 41, 44 are irrelevant and 45 contains y. 38, 43 contain $(M-1)_0$, r_0, as in 10.5.

The static coding of the boxes I-X follows:

B.1	a			
2	b			
I,1	B.2		Ac	b
2	B.1	h-	Ac	b-a
3	s.1	S	s.1	b-a
4	45		Ac	y
5	B.1	h-	Ac	y-a
6	s.1	+	R	$\frac{y-a}{b-a}$
7		A	Ac	$\frac{y-a}{b-a}$
8	s.1	S	s.1	$\frac{y-a}{b-a}$
B.3	$(N-1)_0$			
I,9	B.3		Ac	$(N-1)_0$
10	B.3	h	Ac	$2(N-1)_0$
11	s.2	S	s.2	$2(N-1)_0$
12	38		Ac	$(M-1)_0$
13	42	h-	Ac	$(M-2)_0$
14	42	h-	Ac	$(M-3)_0$
15	s.2	+	R	$\frac{M-3}{2(N-1)} = \frac{(M-3)_0}{2(N-1)_0}$
16		A	Ac	$\frac{M-3}{2(N-1)}$
17	s.2	S	s.2	$\frac{M-3}{2(N-1)}$
18	s.1		Ac	$\frac{y-a}{b-a}$
19	s.2	h-	Ac	$v = \frac{y-a}{b-a} - \frac{M-3}{2(N-1)}$
20	s.1	S	s.1	v
21	s.1	R	R	v
22	B.3		Ac	$(N-1)_0$
23	s.2	Sp'	s.2	$u = 2^{-39}(N-1)$
24	s.2	x	Ac	$2^{-39}k^* = \overrightarrow{uv}$
25	C.1	S	C.1	$2^{-39}k^*$
(to II,1)				
B.4	2^{-39}			
II,1	C.1		Ac	$2^{-39}k^*$
2	B.4	h-	Ac	$2^{-39}(k^*-1)$
3	III,1	Cc		
(to IV,1)				

III,1	B.3		Ac	$(N-1)_o$
2	38	h-	Ac	$(N-M)_o$
3	42	h	Ac	$(N-M+1)_o$
4	C.1.1	Sp'	C.1.1	$2^{-39}(N-M+1)$
5	C.1		Ac	$2^{-39}k^*$
6	C.1.1	h-	Ac	$2^{-39}(k^*-N+M-1)$
7	V,1	Cc		
(to VI,1)				
IV,1	B.4		Ac	$2^{-39}\bar{k} = 2^{-39}$
(to V,2)				
V,1	C.1.1		Ac	$2^{-39}\bar{k} = 2^{-39}(N-M+1)$
2	C.1	S	C.1	$2^{-39}\bar{k}$
(to VII,1)				
VI,	---			
(to VII,1)				
VII,1	C.1		Ac	$2^{-39}\bar{k}$
2	s.1	Sp	s.1	$2^{-19}\bar{k}$
3	s.1	h	Ac	$\bar{k}_o = 2^{-19}\bar{k}+2^{-39}\bar{k}$
B.5	$(\bar{p}-1)_o$			
VII,4	B.5	h	Ac	$(\bar{p}+\bar{k}-1)_o$
5	41	S	41	$(\bar{p}+\bar{k}-1)_o$
6	s.1	S	s.1	$(\bar{p}+\bar{k}-1)_o$
B.6	$(q-1)_o$			
VII,7	B.6		Ac	$(q-1)_o$
8	s.1	h-	Ac	$(q-\bar{p}-\bar{k})_o$
9	44	S	44	$(q-\bar{p}-\bar{k})_c$
(to VIII,1)				
VIII,1	B.2		Ac	b
2	B.1	h-	Ac	b-a
3	s.1	S	s.1	b-a
4	42		Ac	1_o
5	B.3	÷	R	$\frac{1}{N-1} = \frac{1_o}{(N-1)_o}$
6	s.1	x	Ac	$\frac{b-a}{N-1}$
7	C.1.1	S	C.1.1	$\frac{b-a}{N-1}$
8	C.1.1	R	R	$\frac{b-a}{N-1}$
9	C.1		Ac	$2^{-39}\bar{k}$
10	B.4	h-	Ac	$2^{-39}(\bar{k}-1)$

VIII,11	s.1	S	s.1	$2^{-39}(\bar{k}-1)$	
12	s.1	x	R	$\frac{\bar{k}-1}{N-1}(b-a)$	
13		A	Ac	$\frac{\bar{k}-1}{N-1}(b-a)$	
14	B.1	h	Ac	$x_1 = a + \frac{\bar{k}-1}{N-1}(b-a)$	
15	C.1	S	C.1	x_1	
16	B.6		Ac	$(q-1)_0$	
17	42	h	Ac	q_0	
18	C.2	S	C.2	q_0	
(to IX,1)					
IX,1	C.2		Ac	$(q+i-1)_0$	
2	IX,6	Sp	IX,6	$q+i-1$	S
3	42	h	Ac	$(q+i)_0$	
4	C.2	S	C.2	$(q+i)_0$	
5	C.1		Ac	x_i	
6	-	S			
[	q+i-1	S]	B'.i	x_i	
7	C.1.1	h	Ac	$x_{i+1} = x_i + \frac{b-a}{N-1}$	
8	C.1	S	C.1	x_{i+1}	
(to X,1)					
X,1	C.2		Ac	$(q+i)_0$	
2	38	h-	Ac	$(q+i-M+1)_0$	
3	B.6	h-	Ac	$(i-M+2)_0$	
4	42	h-	Ac	$(i-M+1)_0$	
5	42	h-	Ac	$(i-M)_0$	
6	e'	Cc			
(to IX,1)					

Note, that the box VI required no coding, hence its immediate successor (VII) must follow directly upon its immediate predecessor (III).

The ordering of the boxes is I, II, IV; III, VII, VIII, IX, X; V and VII, IX must also be the immediate successors of V, X, respectively, and V,2 must be the immediate successor of IV. This necessitates the extra orders

V,3	VII,1	C
X,7	IX,1	C

and

IV,2	V,2	C

As indicated in Figure 10.4, e' is 0.

We must now assign B.1-6, C.1-2, 1.1, s.1-2 their actual values, pair the 82 orders I,1-25, II,1-3, III,1-7, IV,1-2, V,1-3, VII,1-9, VIII,1-18, IX,1-8, X,1-7 to 41 words, and then assign I,1-X,7 their actual values. We wish to do this as a continuation of the code of 10.5. We will therefore begin with the number 53. Furthermore the contents of C.1-2, C.1.1, s.1-2 are irrelevant like those of 48-52 there. Hence they may be made to coincide with these. We therefore identify them accordingly . Summing all these things up, we obtain the following table:

I,1-25	53 -65	VII,1-9	71'-75'	V,1-3	92'-93'
II,1-3	65'-66'	VIII,1-18	76 -84'	B.1-6	94 -99
IV,1-2	67 -67'	IX,1-8	85 -88'	C.1-2	48 -49
III,1-7	68 -71	X,1-7	89 -92	C.1.1	50
				s.1-2	51-52

Now we obtain this coded sequence:

53	95 ,	94 h-	69	42 h ,	50 Sp'	85	49 ,	87 Sp'
54	51 S ,	45	70	48 ,	50 h-	86	42 h ,	49 S
55	94 h-,	51 ÷	71	92 Cc',	48	87	48 ,	- S
56	A ,	51 S	72	51 Sp ,	51 h	88	50 h ,	48 S
57	96 ,	96 h	73	98 h ,	41 S	89	49 ,	38 h-
58	52 S ,	38	74	51 S ,	99	90	99 h-,	42 h-
59	42 h-,	42 h-	75	51 h- ,	44 S	91	42 h-,	0 Cc
60	52 ÷ ,	A	76	95 ,	94 h-	92	85 C ,	50
61	52 S ,	51	77	51 S ,	42	93	48 S ,	71 C'
62	52 h-,	51 S	78	96 ÷ ,	51 x	94	a	
63	51 R ,	96	79	50 S ,	50 R	95	b	
64	52 Sp',	52 x	80	48 ,	97 h-	96	$(N-1)_0$	
65	48 S ,	48	81	51 S ,	51 x	97	2^{-39}	
66	97 h-,	68 Cc	82	A ,	94 h	98	$(\bar{p}-1)_0$	
67	97 ,	93 C	83	48 S ,	99	99	$(q-1)_0$	
68	96 ,	38 h-	84	42 h ,	49 S			

For the sake of completeness, we restate that part of 0-52 of 10.5 which contains all changes and all substitutable constants of the problem. This is 38-52:

38	$(M-1)_0$	43	r_0	48	--
39	5_0	44	--	49	--
40	10_0	45	y	50	--
41	--	46	8_0	51	--
42	1_0	47	35_0	52	--

The durations may be estimated as follows:

I: 1,220 μ, II: 125 μ, III: 275 μ, IV: 150 μ, V: 125 μ, VII: 350 μ, VIII: 915 μ, IX: 300 μ, X: 275 μ.

Total: I + II + (IV or (III + V) or III) + VII + VIII + (IX + X) $\times$ M =

maximum = (1,220 + 125 + 400 + 350 + 915 + 575 M) μ =

= (575 M + 3,010) $\mu \approx$ (.6 M + 3) m.

Hence the complete interpolation procedure (10.5 plus 10.7) requires

$$(.9\ M^2 + .2\ M + 2.4)\ m.$$

10.8 We consider next Problem 13.b. We are again looking for that $k = 1,\ldots,N-M+1$, for which $y_k,\ldots,y_{k+M-1}$ lie as close to y as possible (cf. the end of 10.6), i.e., for which the remoter one of y_k and y_{k+M-1} lies as close to y as possible.

Since $y_1,\ldots,y_N$ are not equidistant, we cannot find k by an arithmetical criterium, as in 10.7. We must proceed by trial and error, and we will now describe a method which achieves this particularly efficiently.

We are then looking for a $k = \bar{k}$ which minimizes $\mu_k = \text{Max}\ (y-y_k,\ y_{k+M-1}-y)$ in $k = 1,\ldots,N-M+1$.

Let us first note this: $y-y_k > y-y_{k+1}$, hence $y-y_k > y_{k+M}-y$ implies $y-y_k > \mu_{k+1}$, and therefore $\mu_k > \mu_{k+1}$. On the other hand $y_{k+M-1}-y < y_{k+M}-y$, hence $y-y_k \leqq y_{k+M}-y$ implies $\mu_k \leqq y_{k+M}-y$, and therefore $\mu_k \leqq \mu_{k+1}$. Hence $\mu_k >$ or $\leqq \mu_{k+1}$, according to whether $y-y_k >$ or $\leqq y_{k+M}-y$ i.e., $y_k + y_{k+M} - 2y <$ or $\geqq 0$.

In order to keep the size between -1 and 1, it is best to replace $y_k + y_{k+M}-y - 2y$ by

$$(1) \qquad z_k = \frac{1}{2}\,y_k + \frac{1}{2}\,y_{k+M} - y \quad .$$

z_k is monotone increasing in k. Therefore $z_k < 0$ implies $z_h < 0$ and hence $\mu_h > \mu_{h+1}$ for all $h \leqq k$, i.e., $\mu_1 > \mu_2 > \ldots > \mu_k > \mu_{k+1}$. Similarly $z_k \geqq 0$ implies $z_h \geqq 0$ and hence $\mu_h \leqq \mu_{h+1}$ for all $h \geqq k$, i.e., $\mu_k \leqq \mu_{k+1} \leqq \ldots \leqq \mu_{N-M} \leqq \mu_{N-M+1}$. From these we may infer

$$(2) \qquad \begin{array}{l} z_k < 0 \text{ implies } \bar{k} > k, \\ z_k \geqq 0 \text{ implies that we can choose } \bar{k} \leqq k. \end{array}$$

Consequently we can obtain $\bar{k}$ by "bracketing" guided by the sign of z_k.

Note that z_k can be formed for $k = 1,\ldots,N-M$ only, but not for $k = N-M+1$. The "bracketing" must begin by testing the sign of z_k for $k = 1$, if it is +, then $\bar{k} = 1$, if it is -, then we continue. Next we test the sign of z_k for $k = N-M$, if it is -, then $\bar{k} = N-M+1$, if it is +, then we continue. In this case we know that $1 < \bar{k} \leqq N-M$. Put $k_1^- = 1$, $k_1^+ = N-M$. Consider more generally the case where we know that $k_i^- < \bar{k} \leqq k_i^+$. This implies $k_i^+-k_i^- \geqq 1$. If $k_i^+-k_i^- = 1$, then $\bar{k} = k_i^+$; if $k_i^+-k_i^- > 1$, then we continue.

In this case we use the function [w] which denotes the largest integer $\leqq$ w. Put

(3) $$k_i^o = \left[\frac{1}{2} (k_i^+ + k_i^-) \right] ,$$

and test the sign of z_k for $k = k_i^o$, if it is + then $\bar{k} \leqq k_i^o$, if it is - then $\bar{k} > k_i^o$. We can therefore put $k_{i+1}^- = k_i^-$, $k_{i+1}^+ = k_i^o$ or $k_{i+1}^- = k_i^o$, $k_{i+1}^+ = k_i^+$, respectively. This completes the inductive step from i to i+1. Sooner or later, say for $i = i_o$, $k_i^+ - k_i^- = 1$ will occur, and the process will terminate.

Note, that $k_{i+1}^+ - k_{i+1}^- \leqq \frac{1}{2}(k_i^+ - k_i^- + 1)$, i.e., $k_i^+ - k_i^- \geqq 2(k_{i+1}^+ - k_{i+1}^-) - 1$, i.e., $k_{i-1}^+ - k_{i-1}^- \geqq 2(k_i^+ - k_i^-) - 1$. For $i = i_o$, however $k_i^+ - k_i^- = 1$, $k_{i-1}^+ - k_{i-1}^- \neq 1$, hence $k_{i-1}^+ - k_{i-1}^- \geqq 2$. Thus $k_{i_o-1}^+ - k_{i_o-1}^- \geqq 2$, $k_{i_o-2}^+ - k_{i_o-2}^- \geqq 3, \ldots, k_1^+ - k_1^- \geqq 2^{i_o-2}+1$. However, $k_1^+ - k_1^- = N-M-1$, therefore,

(4) $$i_o \leqq {}^2\log(N-M-2) + 2.$$

The virtue of this "bracketing" method is, of course, that the number of steps it requires is of the order of ${}^2\log N$, and not, as it would be with most other trial and error methods, of the order of N. (Note, that N is likely to be large compared to M, which alone figures in the estimates of 10.5 and 10.7.)

After $\bar{k}$ has been obtained, we can utilize the routine of Problem 12 to complete the task, just as at the corresponding point of the discussion of Problem 13.a in 10.7. This means that we propose to use the coded sequence 0-52 of 10.5, and that we will adjust the coded sequence which will be formed here, just as in 10.7.

The situation with the constants of Problem 12 is somewhat simpler than it was in 10.7. Again, p, q, and x need be given values which correspond to the new situation. x is clearly our present y. $p, \ldots, p+M-1$ and $q, \ldots, q+M-1$ are the positions of the $p_1, \ldots, p_M$ and $x_1, \ldots, x_M$ of Problem 12, i.e., the positions of our present $q_{\bar{k}}, \ldots, q_{\bar{k}+M-1}$ and $y_{\bar{k}}, \ldots, y_{\bar{k}+M-1}$, i.e., they are $\bar{p}+\bar{k}-1, \ldots, \bar{p}+\bar{k}+M-2$ and $\bar{q}+\bar{k}-1, \ldots, \bar{q}+\bar{k}+M-2$. Hence $p = \bar{p}+\bar{k}-1$, $q = \bar{q}+\bar{k}-1$. The complications in 10.7, connected with q, or, more precisely, with the $y_{\bar{k}}, \ldots, y_{\bar{k}+M-1}$, do not arise here, since Problem 13.b provided for storing the entire sequence $y_1, \ldots, y_N$.

In assigning letters to the various storage areas to be used, it must be remembered, just as at the corresponding point in 10.7, that the coded sequence that we are now developing is to be used in conjunction with (i.e., as a supplement to) the coded sequence of 10.5. As in 10.7, we have to classify the storage areas required by the latter, but this classification now differs somewhat from that one of 10.7: We have the storage areas D-F, which are incorporated in the final enumeration (they are 38-51 of the 0-52 of 10.5); A and B (i.e., $p, \ldots, p+M-1$ and $q, \ldots, q+M-1$), which will be part of our present A and B (i.e., of $\bar{p}, \ldots, \bar{p}+N-1$ and $\bar{q}, \ldots, \bar{q}+N-1$, they will be $\bar{p}+\bar{k}-1, \ldots, \bar{p}+\bar{k}+M-2$ and $\bar{q}+\bar{k}-1, \ldots, \bar{q}+\bar{k}+M-2$); and C (i.e., $r, \ldots, r+M-2$), which may be at any available place. Therefore, we can, in assigning letters to the various storage areas in the present coding, disregard those of 10.5, with the exception of C. Furthermore, there will again be no need to refer here to C (of 10.5), since the coded sequence of 10.5 assumes the area C to be irrelevantly occupied. We will, at any rate, think of the letters which are meant to designate storage areas of the coded sequence of 10.5 as being primed. We can now assign freely unprimed letters to the storage areas of the present coding.

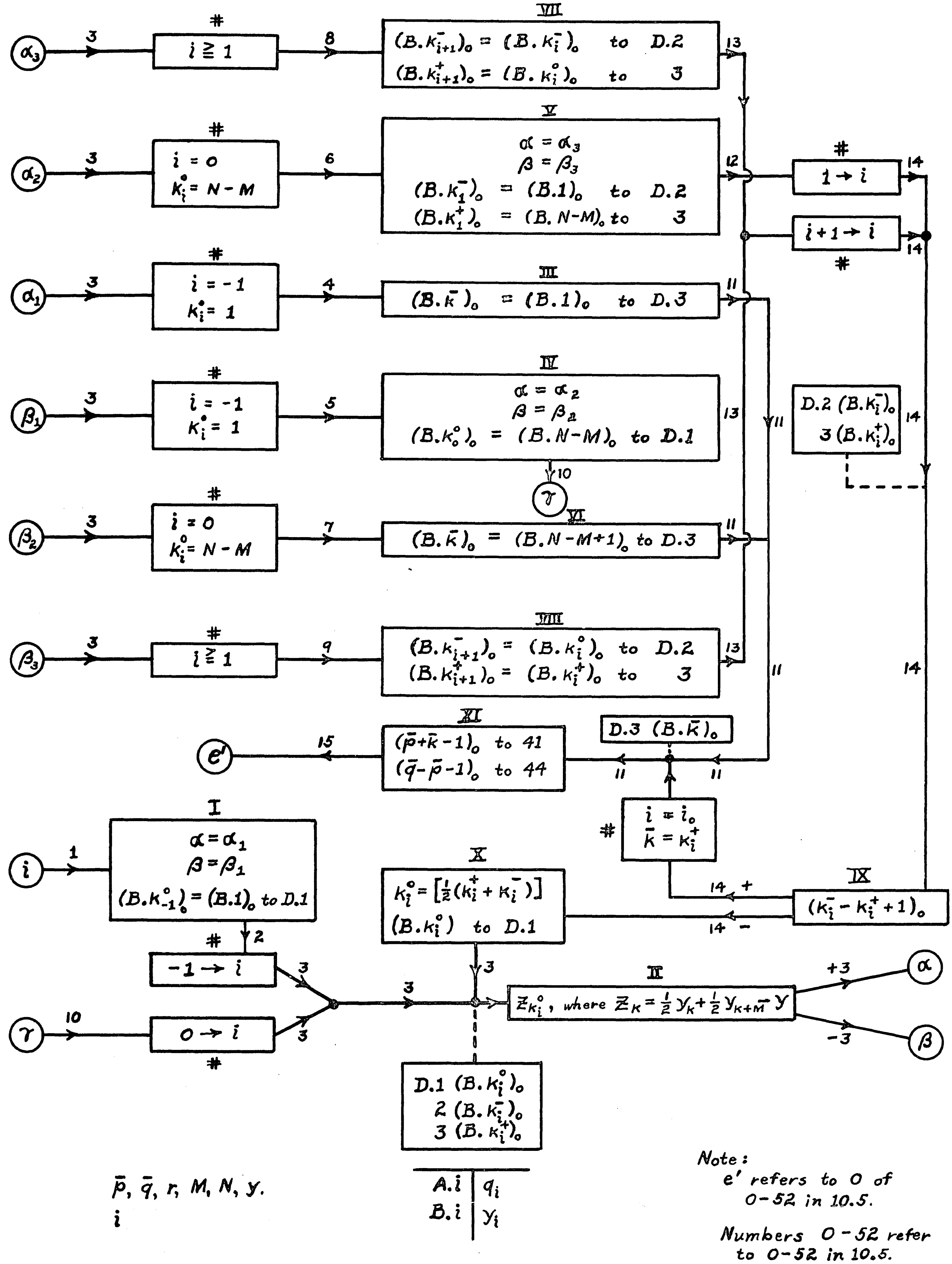
Let A and B be the storage areas corresponding to the intervals from $\bar{p}$ to $\bar{p}+N-1$ and from $\bar{q}$ to $\bar{q}+N-1$. In this way their positions will be A.1,...,N and B.1,...,N, where A.i and B.i correspond to $\bar{p}+i-1$ and $\bar{q}+i-1$ and store q_i and y_i. As in previous problems, the position of A and B need not be shown in the final enumeration of the coded sequence.

The given data of the problem are $\bar{p}$, $\bar{q}$, M, N, y, also r of the coded sequence of 10.5. M, r are already stored there (as $(M-1)_o$, r_o at 38, 43); and y, too (it coincides with x at 45). The definitions of p, q involve $\bar{k}$ ($p=\bar{p}+\bar{k}-1$, $q=\bar{q}+\bar{k}-1$), and $\bar{k}$ originates in the machine. The form in which p is stored accordingly involves $\bar{k}$ (as $p_o = (\bar{p}+\bar{k}-1)_o$ at 41); while the form in which q is stored does not happen to involve $\bar{k}$ (as $(q-p-1)_o = (\bar{q}-\bar{p}-1)_o$ at 44). Hence 41 must be left empty (or rather, irrelevently occupied) when our present coded sequence begins to operate, and it must be appropriately substituted by its operation. 44 might be used to store $(\bar{q}-\bar{p}-1)_o$ from the start, but we prefer to leave it, too, empty (i.e., irrelevantly occupied) and to substitute it in the process. Hence the constants requiring storage now (apart from M, r, y which are stored in 0-52, cf. above) are $\bar{p}$, $\bar{q}$, N. They will be stored in the storage area C. (It will be convenient to store them as $(\bar{p}-1)_o$, $(\bar{q}-1)_o$, $(N-1)_o$.) The exit-locations of the variable remote connections will also be accommodated in C. The induction index i need not be stored explicitly, since the quantities k_i^o, k_i^-, k_i^+ contain all that is needed to keep track of the progress of the induction. (The $i = i_o$ for which the "bracketing", i.e., the induction, ends, is defined by $k_i^+ - k_i^- = 1$, cf. above.) k_i^o, k_i^-, k_i^+ are position marks, they will be stored as $(\bar{q}+k_i^o-1)_o$, $(\bar{q}+k_i^--1)_o$, $(q+k_i^+-1)_o$ (i.e., $(B.k_i^o)_o$, $(B.k_i^-)_o$ $(B.k_i^+)_o$) in the storage area D. Finally, k, when formed (in the form $(\bar{q}+\bar{k}-1)_o$, i.e., $(B.\bar{k})_o$) , will be stored in D, replacing the corresponding expression of k_i^+.

The index i runs from 1 to i_o. In drawing the flow diagram, it is convenient to treat the two preliminary steps (the sensing of the sign of z_k for k = 1 and k = N-M) as part of the same induction, and associate them ideally with two values of i preceding i = 1, i.e., with i = -1 and i = 0. For i = -1, 0 k_i^o may be defined as the value of k for which the sign of z_k is being tested (this is its role for $i \geqq 1$), i.e., as 1, N-M, respectively; while k_i^-, k_i^+ may remain undefined for i = -1, 0.

Our task is to calculate k (in the form $(\bar{q}+\bar{k}-1)_o$, i.e., $(B.\bar{k})_c$); to substitute $p_o = (\bar{p}+\bar{k}-1)_o$ and $(q-p-1)_o = (\bar{q}-\bar{p}-1)_o$ into 41 and 44; and finally to send the control, not to e, but to the beginning of 0-52, i.e. to 0.

We can now draw the flow diagram, as shown in Figure 10.5. The variable remote connections α and β are necessary in order to make the differing treatments for i = -1, for i = 0, and for $i \geqq 1$, possible (cf. above).



VII
(B.k^-_{i+1})$_0$ = (B.k^-_i)$_0$ to D.2
(B.k^+_{i+1})$_0$ = (B.k^o_i)$_0$ to 3
V
α = $α_3$
β = $β_3$
(B.k^-_1)$_0$ = (B.1)$_0$ to D.2
(B.k^+_1)$_0$ = (B. N−M)$_0$ to 3
III
(B.$\bar{k}$)$_0$ = (B.1)$_0$ to D.3
IV
α = $α_2$
β = $β_2$
(B.k^o_0)$_0$ = (B.N−M)$_0$ to D.1
VI
(B.$\bar{k}$)$_0$ = (B.N−M+1)$_0$ to D.3
VIII
(B.k^-_{i+1})$_0$ = (B.k^o_i)$_0$ to D.2
(B.k^+_{i+1})$_0$ = (B.k^+_i)$_0$ to 3
i ≧ 1
i = 0
k^o_i = N − M
i = −1
k^o_i = 1
1 → i
i+1 → i
D.2 (B.k^-_i)$_0$
3 (B.k^+_i)$_0$
D.3 (B.$\bar{k}$)$_0$
XI
($\bar{p}$+$\bar{k}$−1)$_0$ to 41
($\bar{q}$−$\bar{p}$−1)$_0$ to 44
e′
I
α = $α_1$
β = $β_1$
(B.k^o_{-1})$_0$ = (B.1)$_0$ to D.1
i = i_0
$\bar{k}$ = k^+_i
X
k^o_i = [½(k^+_i + k^-_i)]
(B.k^o_i) to D.1
IX
(k^-_i − k^+_i + 1)$_0$
−1 → i
0 → i
II
$Z_{k^o_i}$, where Z_k = ½y_k + ½y_{k+M} − y
D.1 (B.k^o_i)$_0$
2 (B.k^-_i)$_0$
3 (B.k^+_i)$_0$
$\bar{p}$, $\bar{q}$, r, M, N, y.
i
A.i | q_i
B.i | y_i
Note:
e′ refers to 0 of 0−52 in 10.5.
Numbers 0−52 refer to 0−52 in 10.5.

FIGURE 10.5

Regarding the actual coding we make these observations: Since k_i^-, k_i^+ are undefined for i = -1, 0, therefore the contents of D.2, 3 are irrelevant for i = -1, 0 (cf. the storage distributed at the middle bottom of Figure 10.5). The forming of $(B.k_i^o)$ in X, with $k_i^o = [\frac{1}{2}(k_i^+ + k_i^-)]$, is best effected by detouring over the quantities $2^{-39}(\bar{q}+k_i^o-1)$, $2^{-39}(\bar{q}+k_i^--1)$, $2^{-39}(\bar{q}+k_i^+-1)$, because the operation $m = [\frac{1}{2} n]$ is most easily performed with the help of $2^{-39}m$, $2^{-39}n$: Indeed the right-shift R carries the latter directly into the former. The transitions between χ_o and $2^{-39}\chi$ (in both directions) are performed as discussed at the corresponding place in 10.7.

We will use 0-52 in 10.5 just as in 10.7, and the remarks which we made there on this subject apply again.

The static coding of the boxes I-XI follows:

C.1	$(\alpha_1)_o$				
2	$(\beta_1)_o$				
I,1	C.1		Ac	$(\alpha_1)_o$	
2	II,13	Sp	II,13	α_1	Cc
3	C.2		Ac	$(\beta_1)_o$	
4	II,14	Sp	II,14	β_1	C
C.3	$(\bar{q}-1)_o$				
I,5	C.3		Ac	$(\bar{q}-1)_o$	
6	42	h	Ac	$\bar{q}_o$	
7	D.1	S	D.1	$\bar{q}_o$	
(to II,1)					
II,1	D.1		Ac	$(\bar{q}+k_i^o-1)_o$	
2	II,6	Sp	II,6	$\bar{q}+k_i^o-1$	
3	38	h	Ac	$(\bar{q}+k_i^o+M-2)_o$	
4	42	h	Ac	$(\bar{q}+k_i^o+M-1)_o$	
5	II,9	Sp	II,9	$\bar{q}+k_i^o+M-1$	
6	--				
[	$\bar{q}+k_i^o-1$	]	Ac	$y_{k_i^o}$	
7		R	Ac	$\frac{1}{2} y_{k_i^o}$	
8	s.1	S	s.1	$\frac{1}{2} y_{k_i^o}$	
9	--				
[	$\bar{q}+k_i^o+M-1$	]	Ac	$y_{k_i^o+M}$	
10		R	Ac	$\frac{1}{2} y_{k_i^o+M}$	
11	s.1	h	Ac	$\frac{1}{2} y_{k_i^o} + \frac{1}{2} y_{k_i^o+M}$	
12	45	h-	Ac	$z_{k_i^o} = \frac{1}{2} y_{k_i^o} + \frac{1}{2} y_{k_i^o+M} - y$	

II,13	--	Cc				
[	α	Cc	]			
14	--	C				
[	β	C	]			
III,1	C.3			Ac	$(\bar{q}-1)_o$	
2	42	h		Ac	$\bar{q}_o$	
3	D.3	S		D.3	$\bar{q}_o$	
(to XI,1)						
C.4	$(\alpha_2)_o$					
5	$(\beta_2)_o$					
IV,1	C.4			Ac	$(\alpha_2)_o$	
2	II,13	Sp		II,13	α_2	Cc
3	C.5			Ac	$(\beta_2)_o$	
4	II,14	Sp		II,14	β_2	C
C.6	$(N-1)_o$					
IV,5	C.6			Ac	$(N-1)_o$	
6	38	h-		Ac	$(N-M)_o$	
7	C.3	h		Ac	$(\bar{q}+N-M-1)_o$	
8	D.1	S		D.1	$(\bar{q}+N-M-1)_o$	
(to II,1)						
C.7	$(\alpha_3)_o$					
8	$(\beta_3)_o$					
V,1	C.7			Ac	$(\alpha_3)_o$	
2	II,13	Sp		II,13	α_3	Cc
3	C.8			Ac	$(\beta_3)_o$	
4	II,14	Sp		II,14	β_3	C
5	C.3			Ac	$(\bar{q}-1)_o$	
6	42	h		Ac	$\bar{q}_o$	
7	D.2	S		D.2	$\bar{q}_o$	
8	C.3			Ac	$(\bar{q}-1)_o$	
9	C.6	h		Ac	$(\bar{q}+N-2)_o$	
10	38	h-		Ac	$(\bar{q}+N-M-1)_o$	
11	D.3	S		D.3	$(\bar{q}+N-M-1)_0$	
(to IX,1)						
VI,1	C.3			Ac	$(\bar{q}-1)_o$	
2	C.6	h		Ac	$(\bar{q}+N-2)_o$	
3	38	h-		Ac	$(\bar{q}+N-M-1)_o$	
4	42	h		Ac	$(\bar{q}+N-M)_o$	
5	D.3	S		D.3	$(\bar{q}+N-M)_o$	
(to XI,1)						

VII,1	D.1		Ac	$(\bar{q}+k_i^o-1)_0$
2	D.3	S	D.3	$(\bar{q}+k_i^o-1)_0$
(to IX,1)				
VIII,1	D.1		Ac	$(\bar{q}+k_i^o-1)_0$
2	D.2	S	D.2	$(\bar{q}+k_i^o-1)_0$
(to IX,1)				
IX,1	D.2		Ac	$(\bar{q}+k_i^- -1)_0$
2	D.3	h-	Ac	$(k_i^- - k_i^+)_0$
3	42	h	Ac	$(k_i^- - k_i^+ +1)_0$
4	XI,1	Cc		
(to X,1)				
X,1	D.2		Ac	$(\bar{q}+k_i^- -1)_0$
2	D.3	h	Ac	$(2\bar{q}+k_i^+ +k_i^- -2)_0$
3	s.1	Sp'	s.1	$2^{-39}(2\bar{q}+k_i^+ +k_i^- -2)$
4	s.1		Ac	$2^{-39}(2\bar{q}+k_i^+ +k_i^- -2)$
5		R	Ac	$2^{-39}(q+k_i^o-1) =$ $= 2^{-39}(\bar{q}+\frac{1}{2}[k_i^+ +k_i^-] -1)$
6	s.1	Sp	s.1	$2^{-19}(\bar{q}+k_i^o-1)$
7	s.1	h	Ac	$(\bar{q}+k_i^o-1)_0$
8	D.1	S	D.1	$(\bar{q}+k_i^o-1)_0$
(to II,1)				
C.9	$(\bar{p}-1)_0$			
XI,1	C.9		Ac	$(\bar{p}-1)_0$
2	C.3	h-	Ac	$(\bar{p}-\bar{q})_0$
3	D.3	h	Ac	$(\bar{p}+\bar{k}-1)_0$
4	41	S	41	$(\bar{p}+\bar{k}-1)_0$
5	C.3		Ac	$(\bar{q}-1)_0$
6	C.9	h-	Ac	$(\bar{q}-\bar{p})_0$
7	42	h-	Ac	$(\bar{q}-\bar{p}-1)_0$
8	44	S	44	$(\bar{q}-\bar{p}-1)_0$
9	e'	C		

The ordering of the boxes is I, II; III, XI; IV; V, IX, X; VI; VII; VIII, and II, XI, IX, IX, II must also be the immediate successors of IV, VI, VII, VIII, X, respectively. This necessitates the extra orders

IV,9	II,1	C
VI,6	XI,1	C
VII,3	IX,1	C
VIII,3	IX,1	C
X,9	II,1	C

As indicated in Figure 10.5, e' is 0.

α_1, α_2, α_3, β_1, β_2, β_3 correspond to III,1, V,1, VII,1, IV,1, VI,1, VIII,1, respectively. Hence in the final enumeration III,1, V,1, VII,1 must have the same parity and IV,1, VI,1, VIII,1, must have the same parity.

We must now assign C.1-9, D.1-3, s.1 their actual values, pair the 78 orders I,1-7, II,1-14, III,1-3, IV,1-9, V,1-11, VI,1-6, VII,1-3, VIII,1-3, IX,1-4, X,1-9, XI,1-9 to 39 words (actually two dummy orders, necessitated by the adjusting of the parities of V,1 and VIII,1 to those of III,1 and IV,1, respectively, increase this to 40) and then assign I,1-XI,9 their actual values. We wish to do this again as a continuation of the code of 10.5. We will therefore again begin with the number 53. Furthermore the contents of D.1-3, s.1 are irrelevant like those of 48-52 there. Hence they may be made to coincide with four of these. We therefore identify them with 48-51 there. Summing all these things up, we obtain the following table:

I,1-7	53 -56	V,1-11	74'-79'	VIII,1-3	91'-92'
II,1-14	56'-63	IX,1-4	80 -81'	C.1-9	93 -101
III,1-3	63'-64'	X,1-9	82 -86	D.1-3	48 -50
XI,1-9	65 -69	VI,1-6	86'-89	s.1	51
IV,1-9,*	69'-74	VII,1-3,*	89'-91		

Now we obtain this coded sequence:

53	93 ,	62 Sp'	69	0 C ,	96	85	51 h ,	48 S
54	94 ,	63 Sp	70	62 Sp',	97	86	56 C',	95
55	95 ,	42 h	71	63 Sp,	98	87	98 h ,	38 h-
56	48 S ,	48	72	38 h-,	95 h	88	42 h ,	50 S
57	59 Sp,	38 h	73	48 S ,	56 C'	89	65 C ,	48
58	42 h ,	60 Sp'	74	- ,	99	90	50 S ,	80 C
59	- ,	R	75	62 Sp',	100	91	- ,	48
60	51 S ,	-	76	63 Sp,	95	92	49 S ,	80 C
61	R ,	51 h	77	42 h ,	49 S	93	63_0	
62	45 h-,	- Cc'	78	95 ,	98 h	94	69_0	
63	- C',	95	79	38 h-,	50 S	95	$(\bar{q}-1)_0$	
64	42 h ,	50 S	80	49 ,	50 h-	96	74_0	
65	101 ,	95 h-	81	42 h ,	65 Cc	97	86_0	
66	50 h ,	41 S	82	49 ,	50 h	98	$(N-1)_0$	
67	95 ,	101 h-	83	51 Sp',	51	99	89_0	
68	42 h-,	44 S	84	R ,	51 Sp	100	91_0	
						101	$(\bar{p}-1)_0$	

The state of 0-52 of 10.5 must be exactly as described at the end of 10.7.

The durations may be estimated as follows:

I: 275 μ, II: 485 μ , III: 125 μ , IV: 350 μ , V: 425 μ , VI: 225 μ , VII: 125 μ , VIII: 125 μ , IX: 150 μ , X: 330 μ , XI: 350 μ.

Total: $I + [II \text{ or } II \times 2 \text{ or } II \times (i_o+1)] + [III \text{ or } (IV + VI) \text{ or } \{(IV + V +$

$+ (VII \text{ or } VIII) \times (i_o-1) + IX \times i_o + X \times (i_o-1)\}] + XI$

(the two first alternatives in each bracket [ ] refer to two possibilities which can be described by $i_o = 1$ and $i_o = 0$)

$$\text{maximum} = (275 + 485\,(\bar{i}_o+1) + (775 + 125\,(i_o-1) + 150\,i_o + 330\,(i_o-1) + 350)\,\mu =$$

$$= (1{,}090\; i_o + 1{,}430)\;\mu.$$

$$\text{maximum} = (1{,}090\; {}^2\log\,(N\text{-}M\text{-}2) + 3610)\;\mu \approx (1.1\; {}^2\log\,(N\text{-}M\text{-}2) + 3.6)\;m.$$

Hence the complete interpolation procedure (10.5 plus 10.8) requires

$$(.9\,M^2 - .4\,M + 1.1\; {}^2\log\,(N\text{-}M\text{-}2) + 3)\;m.$$

10.9 To conclude, we take up a variant of the group of Problems 12-13, which illustrates the way in which minor changes in the organization of a problem can be effected ex post, i.e., after the problem has been coded on a different basis. (In this connection, cf. also the discussion of 8.9.)

The description of the function under consideration in Problems 12 and 13.b, i.e., the two sequences $p_1,\ldots,p_M$ and $x_1,\ldots,x_M$ on one hand, and $q_1,\ldots,q_N$ and $y_1,\ldots,y_N$ on the other, were stored in two separate systems of consecutive memory locations: $p,\ldots,p+M-1$ and $q,\ldots,q+M-1$ on one hand, and $\bar{p},\ldots,\bar{p}+N-1$ and $\bar{q},\ldots,\bar{q}+N-1$ on the other. (These are the two storage areas A and B of those two problems.) For Problem 13.a the situation is different, since here $q_1,\ldots,q_N$ alone are stored (at the locations $\bar{p},\ldots,\bar{p}+N-1$, storage area A).

Assume now, that these data are stored together with the p_i, x_i, or the q_i, y_i, alternating. I.e., at the locations $p,\ldots,p+2M-1$, or $\bar{p},\ldots,\bar{p}+2N-1$, with p_i, x_i at $p+2i-2$, $p+2i-1$, or q_i, y_i at $\bar{p}+2i-2$, $\bar{p}+2i-1$.

Since this variant cannot arise for Problem 13.a (cf. above), and since its discussion for Problem 13.b comprises the same for Problem 12 (because the coding of the former requires combining with the coding of the latter, cf. 10.8), therefore we will discuss it for Problem 13.b.

We state accordingly:

PROBLEM 13.c.

Same as Problem 13 in its form 13.b, but the quantities $y_1,\ldots,y_N$ and $q_1,\ldots,q_N$ are stored in one system of 2N consecutive memory locations: $\bar{p}, \bar{p}+1,\ldots,\bar{p}+2N-1$ in this order: $q_1, y_1,\ldots,q_N, y_N$. ----

We must analyze the coded sequences 0-52 of 10.5 and 53-101 of 10.8 which correspond to Problems 12 and 13.b, and determine what changes are necessitated by the new formulation, i.e., by Problem 13.c.

Let us consider the code of 10.8, i.e., of Problem 13.b, first. The changes here are due to two causes: First, y_i is now stored at $\bar{p}+2i-1$ instead of $\bar{q}+i-1$; second, since the code of 10.5, i.e., of Problem 12, will also undergo changes, the substitutions into it will change.

The first group is most simply handled in this way: Assume that $\bar{p}$ is odd. Replace the position marks $(B.i)_0$, which should be $(\bar{p}+2i-1)_0$ instead of $(\bar{q}+i-1)_0$, by their half values: $(\frac{1}{2}(\bar{p}+2i-1))_0 = (\frac{\bar{p}+1}{2} + i-1)_0$. We note, for later use, that this has the effect that D.3 will contain immediately before XI $(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$ instead of $(B.\bar{k})_0$.

Returning to the general question: we must change the coding of all those places, where such a position mark is used to obtain the corresponding y_i. The values of these position marks must be doubled before they are used in this way. Apart from this, however, we must only see to it that $\bar{q}$ is replaced by $\frac{\bar{p}+1}{2}$. This affects C.3 (of 10.8), i.e., 95.

The use of position marks in the above sense occurs only in II, when y_k and y_{k+M} are obtained. This takes place at II,6 and II,9. However, the position marks in question originate in the substitutions II,2 and II,5, and it is therefore at these places that the changes have to apply.

We now formulate an adequate coding to replace II.2-4, so as to give II,2 and II,5 the desired effect. Note that at this point D.1 contains $(\frac{1}{2}(\bar{p}+2k_i^0-1)) = (\frac{\bar{p}+1}{2} + k_i^0-1)_0$ instead of $(\bar{q}+k_i^0-1)_0 = (B.k_i^0)_0$. Hence after II,1 the accumulator contains $(\frac{1}{2}(\bar{p}+2k_i^0-1))_0$.

II,1.1		L	Ac	$(\bar{p}+2k_i^o-1)_o$
2	II,6	Sp	II,6	$\bar{p}+2k_i^o-1$
3	38	h	Ac	$(\bar{p}+2k_i^o+M-2)_o$
4	38	h	Ac	$(\bar{p}+2k_i^o+2M-3)_o$
5	42	h	Ac	$(\bar{p}+2k_i^o+2M-2)_o$
6	42	h	Ac	$(\bar{p}+2k_i^o+2M-1)_o$

Thus we must replace the three orders II,2-4 by the six orders II,1.1-6. Hence the third and second remarks of 8.9 apply: We replace II,2-3 by II,1.1-2, then II,4 by

II,1.2,1	II,1.3	C	

and let II,1.3-6 be followed by

II,1.7	II,5	C	

Then II,1.3-7 can be placed at the end of the entire coded sequence. We will give a final enumeration that expresses these changes after we have performed all the changes that are necessary.

The second group must be considered in conjunction with the code of 10.5, i.e., of Problem 12. In the code of 10.8 only XI is involved in the operations of this group. In the code of 10.5 the situation is as follows: Replace the position marks A.i, B.i (of 10.5), which should be $(p+2i-2)_o$, $(p+2i-1)_o$, i.e., $(\bar{p}+2\bar{k}+2i-4)_o$, $(\bar{p}+2\bar{k}+2i-3)_o$, instead of $(p+i-1)_o$, $(q+i-1)_o$, i.e., $(\bar{p}+\bar{k}+i-2)_o$, $(\bar{q}+\bar{k}+i-2)_o$, as nearly by their half values as possible. Since $\bar{p}$ is odd, we will use the half value of the second expression: $(\frac{1}{2}(\bar{p}+2\bar{k}+2i-3))_o = (\frac{\bar{p}+1}{2}+\bar{k}+i-2)_o$. Then we must change the coding of all those places where such position marks are used to obtain the corresponding p_i and y_i: The values of these position marks must be doubled before they are used for a y_i, and then decreased by 1 before they are used for a p_i. We must also take care that the storage locations, which supply the modified $(A.i)_o$, $(B.i)_o$ position marks, are properly supplied themselves. Apart from these, however, we must only see to it that XI (of 10.8) produces $(\frac{1}{2}(\bar{p}+2k-1))_o = (\frac{\bar{p}+1}{2}+\bar{k}+i-2)_o\big|_{i=1}$ instead of $(\bar{p}+\bar{k}-1)_o = (\bar{p}+\bar{k}+i-2)_o\big|_{i=1}$. Since p, q, or $\bar{p}$, $\bar{q}$, no longer appear independently, it is not necessary to produce $(\bar{q}-\bar{p}-1)_o$ in XI.

The use of position marks in the above sense occurs only in III, V, and VII (of 10.5), when p_1, p_i, and x_i, x_{i+h} are obtained. These take place at III,3, V,2, and VII,14,15,21. However, the position marks in question originate in the substitutions III,2, V,1, and VII,8,9,13, and it is therefore at these places that the changes have to apply.

The position marks $(A.i)_0$, $(B.i)_0$ are supplied from E.2, which in turn is supplied by II and IX (of 10.5). II will now supply to E.2 $(\frac{1}{2}(\bar{p}+2\bar{k}+1))_0 = (\frac{\bar{p}+1}{2} + \bar{k} + i-2)_0\big|_{i=2}$ instead of $(p+1)_0 = (\bar{p}+\bar{k})_0 = (\bar{p}+\bar{k}+i-2)_0\big|_{i=2}$. The function of IX is actually performed by VIII, and it will maintain in E.2 $(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1))_0 = (\frac{\bar{p}+1}{2} + \bar{k} + i-1)_0$ instead of $(p+i(1)_0 = (\bar{p}+\bar{k}+i-1)_0$. If we keep these facts in mind when modifying III, V, VII, then we need not change II, IX (i.e., VIII).

We now give adequate codings, to insert before III,2 and before V,1, and also to replace VII,7 before VII,8,9, and to replace VII,10-12 before VII,13, so as to give III,2, V,1, and VII,8,9,13 the desired effect. Note that at these points, as we observed above, D.4 (i.e., 41, substituted from XI of 10.8) contains $(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$, and E.2 contains $(\frac{1}{2}(\bar{p}+2\bar{k}+1))_0$ and $(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1))_0$, respectively. Hence before III,2, V,1, and VII,7, the accumulator contains $(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$, $(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1))_0$ and $(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1))_0$, respectively.

III,1.1		L	Ac	$(\bar{p}+2\bar{k}-1)_0$
2	D.5	h-	Ac	$(\bar{p}+2\bar{k}-2)_0$
V,0.1		L	Ac	$(\bar{p}+2\bar{k}+2i-1)_0$
2	D.5	h-	Ac	$(\bar{p}+2\bar{k}+2i-2)_0$
VII,6.1	D.5	h-	Ac	$(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1)-1)_0$
2		L	Ac	$(\bar{p}+2\bar{k}+2i-3)_0$
VII,9.1	E.2		Ac	$(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1))_0$
2	D.1	h	Ac	$(\frac{1}{2}(\bar{p}+2\bar{k}-2i-1)+M-1)_0$
3	E.1	h-	Ac	$(\frac{1}{2}(\bar{p}+2\bar{k}+2i-1)+h-1)_0$
4		L	Ac	$(\bar{p}+2\bar{k}+2i+2h-3)_0$

Thus we must insert the orders III,1.1-2 before III,2; insert the orders V,0.1-2 before V,1; replace the order VII,7 by the two orders VII,6.1-2; and replace the three orders VII,10-12 by the four orders VII,9.1-4. Hence the third and second remarks of 8.9 apply again: We replace III,1 by

III,0	III,1.0	C	

and let III,1.0 coincide with III,1 and be followed by III,1.1-2 and then by

III,1.3	III,2	C	

Next we replace V,1 by

V,0	V,0.1	C	

and let V,0.1 be as specified above, and be followed by V,0.2, then by V,1, and finally by

V,1.1	V,2	C	

Finally we replace VII,7 and VII,10-12 by VII,6.1-2 and VII,9.1-4. It is best to effect this as a replacement of the entire piece VII,7-12 by VII,6.1-2, VII,8-9, VII,9.1-4 (a total of six orders to be replaced by a total of eight orders). This means that we replace VII,7-11 by VII,6.1-2, VII,8-9, VII,9.1, then we replace VII,12 by

VII,9.1.1	VII,9.2	C	

and let VII,9.2-4 be followed by

VII,9.2.5	VII,13	C	

Then III,1.0-3; V,0.1-2, V,1, V,1.1; VII,9.2-5 can be placed at the end of the entire coded sequence. We will give a final enumeration that expresses these changes after we have performed all the changes that are necessary.

Returning to the second group in the coding of 10.8, we note that it affects only XI (of 10.8). We saw above that XI must produce $(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$ and substitute it into 41, and that this is all that has to be done there. However, we noted before that D.3 before XI contains precisely $(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$. Hence XI,1-9 (i.e., XI in its entirety) may be replaced by

XI,0.1	D.3		Ac	$(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$
2	41	S	41	$(\frac{1}{2}(\bar{p}+2\bar{k}-1))_0$
3	e'	C		

Hence we replace XI,1-3 by XI,0.1-3, and since operations of the code of 10.8 end at this point, XI,4-9 may be left empty.

To conclude, we observe that C.3 must contain $(\frac{\bar{p}-1}{2})_0 = (\frac{\bar{p}+1}{2} - 1)_0$ instead of $(\bar{q}-1)_0$, and that neither $\bar{p}$ nor $\bar{q}$ are needed in any other form, so that C.9 may be left empty.

We are now in possession of a complete list of all changes of both codes 0-52 of 10.5 and 53-101 of 10.8. We tabulate the omissions and modifications first, and the additions afterwards.

Omissions and Modifications:

From		To
(10.8) :	II,2-3	II,1.1-2
	II,4	II,1.2.1
(10.5) :	III,1	III,0
	V,1	V,0
	VII,7-8	VII,6.1-2
	VII,9-10	VII,8-9
	VII,11	VII,9.1
	VII,12	VII,9.1.1

Omissions and Modifications (cont.):

	From	To
(10.8) :	XI,1-3	XI,0.1-3
	XI,4-9	Empty
	C.3	$(\frac{p-1}{2})_0$
	C.9	Empty

Additions:

(10.8) :	II,1.3-7
(10.5) :	III,1.0-3
(10.5) :	V,0.1-2
	V,1
	V,1.1
(10.5) :	VII,9.2-5

We can use five of the six empty fields XI,4-9 (from 10.8) to accommodate II,1.3-7 (from 10.8) -- say XI,4-8. C.9 is the last word of the code: 101, hence we can place III,1.0-3; V,0.1-2, V,1, V,1.1, VII,9.2-5 (from 10.5) in a sequence that begins there. They will then occupy these positions:

III,1.0-3	101-102'	V,1	104	VII,2-5	105-106'
V,0.1-2	103-103'	V,1.1	104'		

(All these are from 10.5.)

Now we can formulate the final form of all changes in the coded sequences 0-52 and 53-101:

5	101 C ,		58	66 C' ,		95	$(\frac{p-1}{2})_0$	
10	103 C ,		65	50 ,	41 S	101	41 ,	L
14		42 h-	66	e'C ,	38 h	102	42 h-,	5 C'
15	L ,	18 Sp'	67	38 h ,	42 h	103	L ,	42 h-
16	21 Sp',	49	68	42 h ,	58 C'	104	10 Sp',	10 C'
17	105 C ,		69	-- ,		105	38 h ,	48 h-
57	L ,	59 Sp				106	L ,	17 C'

The following durations are affected:

(10.5) :	III: + 130 μ
	V: + 130 μ
	VII: + 130 μ
(10.8) :	II: + 180 μ
	IX: - 225 μ

Total: (10.5) : $(130 + 130\,(M-1) + 130\,\frac{M(M-1)}{2})\,\mu =$

$= (65\,M^2 + 65\,M)\,\mu \approx (.07\,M^2 + .07\,M)$ m.

Total: (10.8) : $(180\,(i_o + 1) - 225\,i_o)\,\mu = (-\,45\,i_o + 180)\,\mu$

maximum $\approx$ .2 m.

Hence there is no relevant change in the estimate at the end of 10.8.

11.0 CODING OF SOME COMBINATORIAL (SORTING) PROBLEMS

11.1 In this chapter we consider problems of a combinatorial, and not analytical character. This means, that the properly calculational (arithmetical) parts of the procedure will be very simple (indeed almost absent), and the essential operations will be of a logical character. Such problems are of a not inconsiderable practical importance, since they include the category usually referred to as *sorting problems*. They are, furthermore, of a conceptual interest, for the following reason.

Any computing machine is organized around three vital organs: The memory, the logical control, and the arithmetical organ. The last mentioned organ is an adder-subtractor-multiplier-divider, and the multiplier is usually that aspect which primarily controls its speed (cf. the discussions of Part I.) The efficiency of the machine on all analytical problems is clearly dependent to a large extent on the efficiency of the arithmetical organ, and thus primarily of the multiplier. Now the non-analytical, combinatorial problems, to which we have referred, avoid the multiplier almost completely. (Adding and subtracting is hardly avoidable even in purely logical procedures, since the operations are needed to construct the position marks of memory locations and to effect size comparisons of numbers, by which we have to express our logical alternatives.) Consequently these problems provide tests for the efficiency of the non-arithmetical parts of the machine: The memory and the logical control.

11.2 We will consider two typical sorting problems: Those of *meshing* and of *sorting* proper. There are, of course, many others, and many variants for each problem, but these two should suffice to illustrate the main methodical principles.

In order to formulate our two problems, we need the definitions which follow.

We operate with *sequences* of *complexes*. A complex $X = (x;\ u_1,\dots,u_p)$ consists of p+1 numbers: The *principal number* x and the *subsidiary numbers* $u_1,\dots,u_p$. p is the *order* of the complex. A *sequence* $S = (X^{(1)},\dots,X^{(n)})$ consists of n complexes $X^{(i)} = (x^{(i)};\ u_1^{(i)},\dots,u_p^{(i)})$. n is the *length* of the sequence. Throughout what follows we will never vary p, the (arbitrary but) fixed order of all complexes to be considered. We will, however, deal with sequences of various lengths.

A sequence $S = (X^{(1)},\dots,X^{(n)})$ is a *monotone* if the principal elements $x^{(i)}$ of its complexes $X^{(i)} = (x^{(i)};\ u_1^{(i)},\dots,u_p^{(i)})$ form a monotone, non decreasing sequence:

$$x^{(1)} \leqq x^{(2)} \leqq \dots \leqq x^{(n)} \tag{1}$$

A sequence $T = (Y^{(1)},\dots,Y^{(n)})$ is a *permutation* of another sequence $S = (X^{(1)},\dots,X^{(n)})$ of the same length, if for a suitable permutation $i \rightarrow i'$ of the $i = 1,\dots,n$

$$Y^{(i)} = X^{(i')} \qquad (i = 1,\dots,n). \tag{2}$$

Given two sequences $S = (X^{(1)},\ldots,X^{(n)})$, $T = (Y^{(1)},\ldots,Y^{(m)})$ (of not necessarily equal lengths n, m), their *sum* is the sequence $[S, T] = (X^{(1)},\ldots,X^{(n)}, Y^{(1)},\ldots,Y^{(m)})$ (of length n + m).

We can now state our two problems:

PROBLEM 14. (Meshing)

Two monotone sequences S, T, of lengths n, m, respectively, are stored at two systems of n(p+1), m(p+1) consecutive memory locations respectively: s, s+1,..., s+n(p+1)-1 and t, t+1,..., t+m(p+1)-1. An accomodation for a sequence of length n + m, consisting of (n + m)(p + 1) consecutive memory locations, is available: r, r+1,..., r+(n+m)(p+1)-1. The constants of the problem are p, n, m, s, t, r, to be stored at six given memory locations. It is desired to find a monotone permutation R of the sum $[S, T]$, and to place it at the locations r, r+1,..., r+(n+m)(p+1)-1. ----

PROBLEM 15. (Sorting)

A sequence S (subject to no requirements of monotony whatever) is stored at n(p+1) consecutive memory locations: s, s+1,..., s+n(p+1)-1. The constants of the problem are s, n, p, to be stored at three given memory locations. It is desired to find a monotone permutation S^* of S, and to replace S by it.

11.3 We consider first Problem 14, which is the simpler one, and whose solution can be conveniently used in solving Problem 15.

We have $S = (X^{(1)},\ldots, X^{(n)})$, $T = (Y^{(1)},\ldots, Y^{(m)})$ with $X^{(i)} = (x^{(i)}; u_1^{(i)},\ldots, u_p^{(i)})$, $Y^{(j)} = (y^{(j)}; v_1^{(j)},\ldots, v_p^{(j)})$, and we wish to form $R = (Z^{(1)},\ldots, Z^{(n+m)})$ with $Z^{(k)} = (z^{(k)}; w_1^{(k)},\ldots,w_p^{(k)})$, which is a monotone permutation of $[S, T]$.

Forming R can be viewed as an inductive process: If $Z^{(1)},\ldots,Z^{(k-1)}$ for a k=1,...,n+m, have already been formed, the inductive step consists of forming $Z^{(k)}$. It is preferable to allow k = n+m+1, too, in the sense that when this value of k is reached, the process is terminated. Assume, therefore, that we have formed $Z^{(1)},\ldots,Z^{(k-1)}$ for a k = 1,...,n+m+1. It is clear, that we may assume in addition, that these $Z^{(1)},\ldots,Z^{(k-1)}$ are the $X^{(1)},\ldots,X^{(i_k)}$ together with the $Y^{(1)},\ldots,Y^{(j_k)}$, for two suitable i_k, j_k with

(1) $$i_k + j_k = k-1 \qquad i_k = 1,\ldots, n, \quad j_k = 1,\ldots, m.$$

Then we have the following possibilities:

(a) $i_k = n$, $j_k = m$: By (1) k = n+m+1, hence, as we observed above, R is fully formed, and nothing remains to be done.

(b) Not $i_k = n$, $j_k = m$: By (1) $k \leqq n+m$, hence, as we observed above, R is not fully formed, and $Z^{(k)}$ must be formed next. Clearly, we may choose $Z^{(k)}$ as one of $X^{(i_k+1)}$ and $Y^{(j_k+1)}$, we must decide which. We introduce a quantity ω_k, so that $\omega_k = 0$ for $Z^{(k)} = X^{(i_k+1)}$ and $\omega_k = 1$ for $Z^{(k)} = Y^{(j_k+1)}$.

$\omega_k = 0$ implies $i_{k+1} = i_k + 1$, $j_{k+1} = j_k$; $\omega_k = 1$ implies $i_{k+1} = i_k$, $j_{k+1} = j_k + 1$. ω_k is determined according to the following rules:

(b.1) $i_k = n$, hence $j_k \neq m$: S is exhausted, hence $\omega_k = 1$.

(b.2) $j_k = m$, hence $i_k \neq n$: T is exhausted, hence $\omega_k = 0$.

(b.3) $i_k \neq n$, $j_k \neq m$: Neither S nor T is exhausted. We must distinguish further:

(b.3.1) $x^{(i_k+1)} < y^{(j_k+1)}$: Necessarily $\omega_k = 0$.

(b.3.2) $x^{(i_k+1)} > y^{(j_k+1)}$: Necessarily $\omega_k = 1$.

(b.3.3) $x^{(i_k+1)} = y^{(j_k+1)}$: Both $\omega_k = 0$ and $\omega_k = 1$ are acceptable. We choose $\omega_k = 0$.

As pointed out above, (a), (b) describes an inductive process which runs over $k = 1, \ldots, n+m+1$. It begins with $k = 1$ and $i_k = j_k = 0$ (owing to (1)), and then progresses from k to $k+1$, forming the necessary i_k, j_k and ω_k as it goes along.

It is convenient to denote the locations s and t, where the storage of the sequences S and T begins, by a common letter and to distinguish them by an index:

(2) $$s^0 = s, \qquad s^1 = t.$$

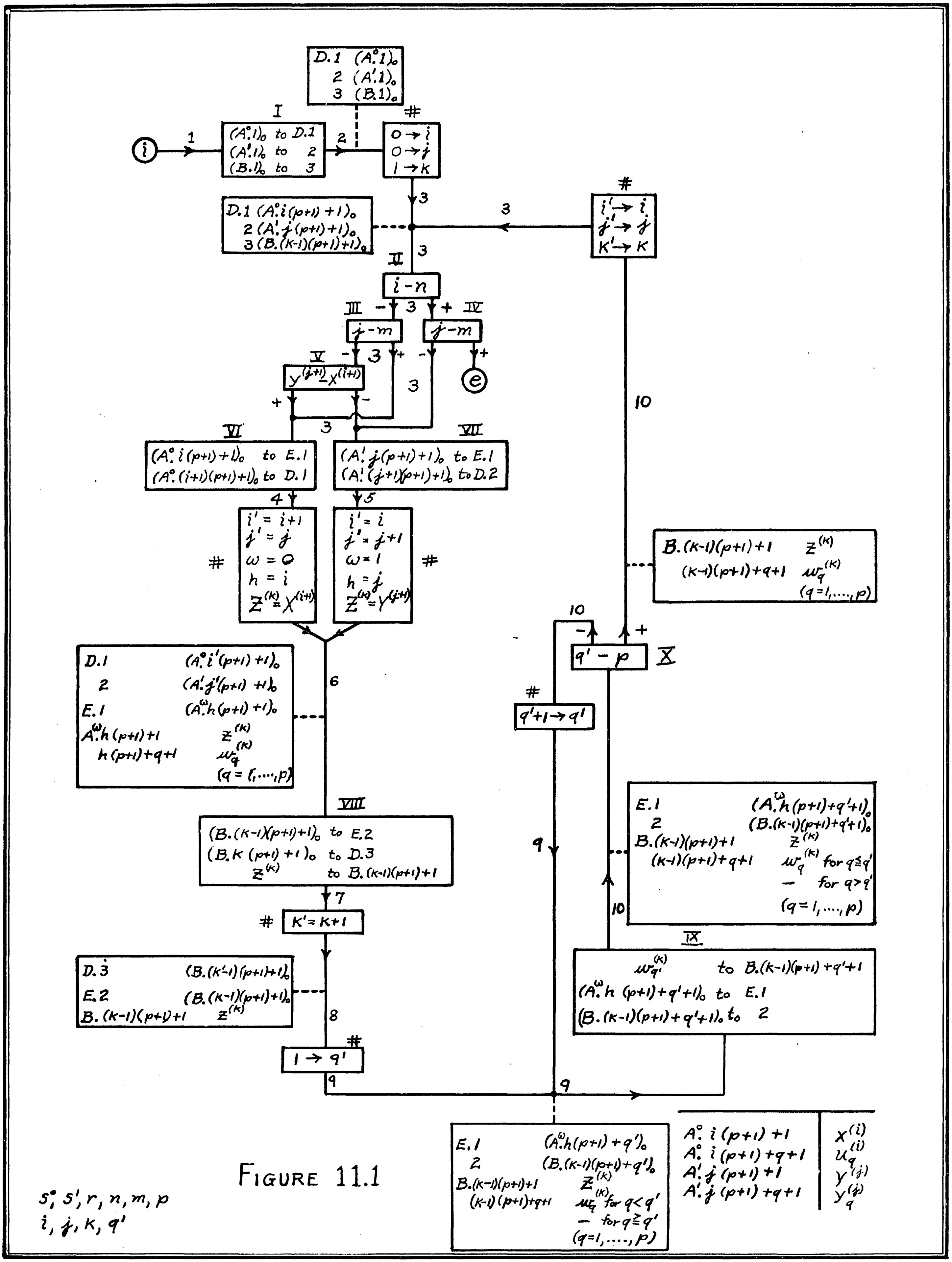
Correspondingly, let A^0 and A^1 be the storage areas corresponding to the two intervals of locations from s^0 to $s^0 + n(p+1)-1$ and from s^1 to $s^1 + m(p+1)-1$. In this way their positions will be $A^0.1, \ldots, n(p+1)$ and $A^1.1, \ldots, m(p+1)$, where $A^0.a$ and $A^1.b$ correspond to $s^0 + a-1$ and $s^1 + b-1$, and store $x^{(i)}$ for $a = (i-1)(p+1)+1$, $u_q^{(i)}$ for $a = (i-1)(p+1)+q+1$, $y^{(j)}$ for $b = (j-1)(p+1)+1$, $v_q^{(j)}$ for $b = (j-1)(p+1)+q+1$, respectively. Next, let B be the storage area intended for the sequence R, corresponding to the interval of locations from r to $r+(n+m)(p+1)-1$. In this way its positions will be $B.1, \ldots, (n+m)(p+1)$, where $B.c$ corresponds to $r+c-1$, and is intended to store $z^{(k)}$ for $c = (k-1)(p+1)+1$, $w_q^{(k)}$ for $c = (k-1)(p+1)+q+1$, respectively. As in several previous problems, the positions of A^0, A^1, and B need not be shown in the final enumeration of the coded sequence.

Further storage capacities are needed as follows: The given data (the constants) of the problem, s^0, s^1, r, n, m, p, will be stored in the storage area C. (It will be convenient to store them as $(s^0)_0$, $(s^1)_0$, r_0, $(n(p+1))_0$, $(m(p+1))_0$, $(p+1)_0$. We will also store 1_0 in C.) Next the induction indices k, q' will have to be stored. It is convenient to store i_k, j_k, too, along with k, while it turns out to be unnecessary to form and to store ω_k explicitly. These indices i, j, k, q' are all relevant as position marks, and it will prove convenient to store in their place $(s^0 + i(p+1))_0$, $(s^1 + j(p+1))_0$, $(r + (k-1)(p+1))_0$ $(s^\omega + h(p+1) + q'-1)_0$, $(r + (k-1)(p+1) + q'-1)_0$ (i.e. $(A^0.i(p+1)+1)_0$, $(A^1.j(p+1)+1)_0$, $(B.(k-1)(p+1)+1)_0$, $(A^\omega.h(p+1) + q')_0$, $(B.(k-1)(p+1) + q')_0$; here $\omega = \omega_k$, and $h = i$ or j for $\omega = 0$ or 1, respectively) Note, that two position marks that correspond to q' are being stored. The three first quantities will be stored in the storage area D, and the two last ones in the storage area E.

We can now draw the flow diagram as shown in Figure 11.1. The actual coding obtains from this without a need for further comments.

The static coding of the boxes I-X follows:

C.1	$(s^0)_0$					
I,1	C.1			Ac	$(s^0)_0$	
2	D.1	S		D.1	$(s^0)_0$	
C.2	$(s^1)_0$					
I,3	C.2			Ac	$(s^1)_0$	
4	D.2	S		D.2	$(s^1)_0$	
C.3	r_0					
I,5	C.3			Ac	$(r)_0$	
6	D.3	S		D.3	$(r)_0$	
(to II,1)						
C.4	$(n(p+1))_0$					
II,1	D.1			Ac	$(s^0+i(p+1))_0$	
2	C.1	h-		Ac	$(i(p+1))_0$	
3	C.4	h-		Ac	$((i-n)(p+1))_0$	
4	IV,1	Cc				
(to III,1)						
C.5	$(m(p+1))_0$					
III,1	D.2			Ac	$(s^1+j(p+1))_0$	
2	C.2	h-		Ac	$(j(p+1))_0$	
3	C.5	h-		Ac	$((j-m)(p+1))_0$	
4	VI,1	Cc				
(to V,1)						
IV,1	D.2			Ac	$(s^1+j(p+1))_0$	
2	C.2	h-		Ac	$(j(p+1))_0$	
3	C.5	h-		Ac	$((j-m)(p+1))_0$	
4	e	Cc				
(to VII,1)						
V,1	D.1			Ac	$(s^0+i(p+1))_0$	
2	V,6	Sp		V,6	$s^0+i(p+1)$	h-
3	D.2			Ac	$(s^1+j(p+1))_0$	
4	V,5	Sp		V,5	$s^1+j(p+1)$	
5	--					
[	$s^1+j(p+1)$		]	Ac	y_{j+1}	
6	--	h-				
[	$s^0+i(p+1)$	h-	]	Ac	$y_{j+1}-x_{i+1}$	
7	VI,1	Cc				
(to VII,1)						



D.1 (A°.1)₀
2 (A'.1)₀
3 (B.1)₀
I
(A°.1)₀ to D.1
(A'.1)₀ to 2
(B.1)₀ to 3
0 → i
0 → j
1 → K
D.1 (A°.i(p+1)+1)₀
2 (A'.j(p+1)+1)₀
3 (B.(K-1)(p+1)+1)₀
i' → i
j' → j
K' → K
II
i - n
III
IV
j - m
j - m
V
y^(j+1) - x^(i+1)
VI
(A°.i(p+1)+1)₀ to E.1
(A°.(i+1)(p+1)+1)₀ to D.1
VII
(A'.j(p+1)+1)₀ to E.1
(A'.(j+1)(p+1)+1)₀ to D.2
i' = i+1
j' = j
ω = 0
h = i
z^(K) = x^(i+1)
i' = i
j' = j+1
ω = 1
h = j
z^(K) = y^(j+1)
B.(K-1)(p+1)+1 z^(K)
(K-1)(p+1)+q+1 w_q^(K)
(q = 1, ..., p)
D.1 (A°.i'(p+1)+1)₀
2 (A'.j'(p+1)+1)₀
E.1 (A^ω.h(p+1)+1)₀
A^ω.h(p+1)+1 z^(K)
h(p+1)+q+1 w_q^(K)
(q = 1, ..., p)
q' - p
X
q'+1 → q'
VIII
(B.(K-1)(p+1)+1)₀ to E.2
(B.K(p+1)+1)₀ to D.3
z^(K) to B.(K-1)(p+1)+1
E.1 (A^ω.h(p+1)+q'+1)₀
2 (B.(K-1)(p+1)+q'+1)₀
B.(K-1)(p+1)+1 z^(K)
(K-1)(p+1)+q+1 w_q^(K) for q ≤ q'
— for q > q'
(q = 1, ..., p)
K' = K+1
IX
w_q'^(K) to B.(K-1)(p+1)+q'+1
(A^ω.h(p+1)+q'+1)₀ to E.1
(B.(K-1)(p+1)+q'+1)₀ to 2
D.3 (B.(K'-1)(p+1)+1)₀
E.2 (B.(K-1)(p+1)+1)₀
B.(K-1)(p+1)+1 z^(K)
1 → q'
E.1 (A^ω.h(p+1)+q')₀
2 (B.(K-1)(p+1)+q')₀
B.(K-1)(p+1)+1 z^(K)
(K-1)(p+1)+q+1 w_q^(K) for q < q'
— for q ≥ q'
(q = 1, ..., p)
A°. i(p+1)+1 x^(i)
A°. i(p+1)+q+1 u_q^(i)
A'. j(p+1)+1 y^(j)
A'. j(p+1)+q+1 y_q^(j)
s°, s', r, n, m, p
i, j, K, q'

FIGURE 11.1

C.6	$(p+1)_0$				
VI,1	D.1		Ac	$(s^0+i(p+1))_0$	
2	E.1	S	E.1	$(s^0+i(p+1))_0$	
3	C.6	h	Ac	$(s^0+(i+1)(p+1))_0$	
4	D.1	S	D.1	$(s^0+(i+1)(p+1))_0$	
(to VIII,1)					
VII,1	D.2		Ac	$(s^1+j(p+1))_0$	
2	E.1	S	E.1	$(s^1+j(p+1))_0$	
3	C.6	h	Ac	$(s^1+(j+1)(p+1))_0$	
4	D.2	S	D.2	$(s^1+(j+1)(p+1))_0$	
(to VIII,1)					
VIII,1	D.3		Ac	$(r+(k-1)(p+1))_0$	
2	VIII,9	Sp	VIII,9	$r+(k-1)(p+1)$	S
3	E.2	S	E.2	$(r+(k-1)(p+1))_0$	
4	C.6	h	Ac	$(r+k(p+1))_0$	
5	D.3	S	D.3	$(r+k(p+1))_0$	
6	E.1		Ac	$(s^{(\omega)}+h(p+1))_0$	
7	VIII,8	Sp	VIII,8	$s^{(\omega)}+h(p+1)$	
8	--				
[	$s^{(\omega)}+h(p+1)$	]	Ac	$z^{(k)}$	
9	--	S			
[	$r+(k-1)(p+1)$	S]	B.(k-1)(p+1)+1	$z^{(k)}$	
(to IX,1)					
C.7	1_0				
IX,1	E.2		Ac	$(r+(k-1)(p+1)+q'-1)_0$	
2	C.7	h	Ac	$(r+(k-1)(p+1)+q')_0$	
3	IX,10	Sp	IX,10	$r+(k-1)(p+1)+q'$	S
4	E.2	S	E.2	$(r+(k-1)(p+1)+q')_0$	
5	E.1		Ac	$(s^{(\omega)}+h(p+1)+q'-1)_0$	
6	C.7	h	Ac	$(s^{(\omega)}+h(p+1)+q')_0$	
7	IX,9	Sp	IX,9	$s^{(\omega)}+h(p+1)+q'$	
8	E.1	S	E.1	$(s^{(\omega)}+h(p+1)+q')_0$	
9	--				
[	$s^{(\omega)}+h(p+1)+q'$	]	Ac	$w^{(k)}_{q'}$	
10	--	S			
[	$r+(k-1)(p+1)+q'$	S]	B.(k-1)(p+1)+q'+1	$w^{(k)}_{q'}$	
(to X,1)					

X,1	E.2		Ac	$(r+(k-1)(p+1)+q')_0$
2	D.3	h-	Ac	$(q'-p-1)_0$
3	C.7	h	Ac	$(q'-p)_0$
4	II,1	Cc		
(to IX,1)				

The ordering of the boxes is I, II, III, V, VII, VIII, IX, X, and VII, VIII, IX must also be the immediate successors of IV, VI, X, respectively. This necessitates the extra orders

IV,5	VII,1	C
VI,5	VIII,1	C
X,5	IX,1	C

We must now assign C.1-7, D.1-3, E.1-2 their actual values, pair the 59 orders I,1-6, II,1-4, III,1-4, IV,1-5, V,1-7, VI,1-5, VII,1-4, VIII,1-9, IX,1-10, X,1-5 to 30 words and then assign I,1-X,5 their actual values. These are expressed in this table:

I,1-6	0 -2'	VII,1-4	10'-12	IV,1-5	24'-26'
II,1-4	3 -4'	VIII,1-9	12'-16'	VI,1-5	27 -29
III,1-4	5 -6'	IX,1-10	17 -21'	C.1-7	30 -36
V,1-7	7 -10	X,1-5	22 -24	D.1-3	37 -39
				E.1-2	40 -41

Now we obtain this coded sequence:

0	30 , 37 S	14	35 h , 39 S	28	35 h , 37 S
1	31 , 38 S	15	40 , 16 Sp	29	12 C', - -
2	32 , 39 S	16	- , - S	30	$(s^0)_0$
3	37 , 30 h-	17	41 , 36 h	31	$(s^1)_0$
4	33 h-, 24 Cc'	18	21 Sp', 41 S	32	r_0
5	38 , 31 h-	19	40 , 36 h	33	$(n(p+1))_0$
6	34 h-, 27 Cc	20	21 Sp, 40 S	34	$(m(p+1))_0$
7	37 , 9 Sp'	21	- , - S	35	$(p+1)_0$
8	38 , 9 Sp	22	41 , 39 h-	36	1_0
9	- , - h-	23	36 h , 3 Cc	37	-
10	27 Cc, 38	24	17 C , 38	38	-
11	40 S , 35 h	25	31 h-, 34 h-	39	-
12	38 S , 39	26	e Cc, 10 Cc'	40	-
13	16 Sp', 41 S	27	37 , 40 S	41	-

The durations may be estimated as follows:

I: 225 μ, II: 150 μ, III: 150 μ, IV: 200 μ, V: 275 μ, VI: 200 μ, VII: 150 μ, VIII: 350 μ, IX: 375 μ, X: 200 μ.

$$\text{Total: } I + \left\{ \begin{matrix} II + (III \text{ or } (III + V) \text{ or } IV) + (VI \text{ or } VII) + \\ + VIII + (IX + X) \times p \end{matrix} \right\} \times (n + m) + (II + IV)$$

maximum $= (225 + (150 + 425 + 200 + 350 + 575 \times p) \times (n + m) + 350)\ \mu =$

$= ((575\ p + 1,125)\ (n + m) + 575)\ \mu \approx$

$\approx .6\ ((p + 2)\ (n + m) + 1)\ m$

This result will be analyzed further in 11.5.

11.4 We pass now to Problem 15, which will be solved, as indicated above, with the help of the solution of Problem 14.

We have $S = (X^{(1)}, \ldots, X^{(n)})$ with $X^{(i)} = (x^{(i)};\ u_1^{(i)}, \ldots, u_p^{(i)})$, and we wish to form an $S^* = (X^{(1')}, \ldots, X^{(n')})$, which is a monotone permutation of S.

This can be achieved in the following manner: If a certain permutation $S^{*o} = (X^{(1'^o)}, \ldots, X^{(n'^o)})$ of S has been found, which may not yet be fully monotone, but in which at least two consecutive intervals $X^{(k'^o)}, \ldots, X^{(k+i-1'^o)}$ and $X^{(k+i'^o)}, \ldots, X^{(k+i+j-1'^o)}$ are (each one separately) monotone, then we can mesh these (in the sense of Problem 14), and thereby obtain a new permutation $S^{*oo} = X^{(1'^{oo})}, \ldots, X^{(n'^{oo})})$ of S^{*o}, i.e. of S, for which the whole interval $X^{(k'^{oo})}, \ldots, X^{(k+i+j-1'^{oo})}$ is monotone. Thus we can convert two monotone intervals of the lengths i, j into one of the length i + j. The original S is made up of monotone intervals of length 1, since each $X^{(k)}$ (separately) may be viewed as such an interval. Hence we can, beginning with these monotone intervals of length 1, build up monotone intervals of successively increasing length, until the maximum length n is reached, i.e. the permutation of S at hand is monotone as a whole.

This qualitative description can be formalized to an inductive process. This process goes through the successive stages $v = 0, 1, 2, \ldots$. In the stage v a permutation $S^{*v} = (X^{(1'^v)}, \ldots, X^{(n'^v)})$ of S is at hand, which is made up of consecutive intervals of length 2^v, which are (each one separately) monotone. (Since n may not be divisible by v, the length of the last interval has to be the remainder r_v of the division of n by 2^v, $r_v < 2^v$. If n is divisible by 2^v, we should put $r_v = 0$ and rule this interval to be absent, but we may just as well put $r_v = 2^v$, and still rule it to be the last interval.) Now a sequence of steps of the nature described above will produce a new permutation S^{*v+1} of S^{*v}, i.e. of S, in which the monotone intervals of length 2^v in S^{*v} are pairwise meshed to monotone intervals of length 2^{v+1} in S^{*v+1}. (The end-effect for v + 1 can be described in the same terms as above for v.) This inductive process begins with $v = 0$ and $S^{*0} = S$. It ends when a v with $2^v \geq n$ has been reached, the corresponding S^{*v} is the desired (monotone) S^*. (Note, that at this point there is only one interval: It is the final interval of length $r_v \leqq 2^v$ referred to above, in this case with $r_v = n$.) Using the function $\langle z \rangle$, which denotes the smallest integer $\geqq z$, we may therefore say that this process terminates with $v = \langle {}^2\log n \rangle$.

The induction over $v = 0, 1, 2, \ldots, \langle {}^2\log n \rangle$ is, however, only the primary induction. For each v we have $\left\langle \frac{n}{2^v} \right\rangle$ intervals of length 2^v (cf. above), these form $\left\langle \frac{1}{2} \left\langle \frac{n}{2^v} \right\rangle \right\rangle = \left\langle \frac{n}{2^{v+1}} \right\rangle$ pairs, which have to be meshed to $\left\langle \frac{n}{2^{v+1}} \right\rangle$ intervals of length 2^{v+1} (cf. above), in order to effect the inductive step from v to v + 1.

If we enumerate these interval pairs by an index $w = 1,\ldots,\langle\frac{n}{2^{v+1}}\rangle$, then it becomes clear that we are dealing with a secondary induction, of which w is the index.

We can now state the general inductive step with complete rigor:

Consider a $v = 0, 1, 2,\ldots$, and assume that the permutation $S^{*v} = (X^{(1'^v)},\ldots,X^{(n'^v)})$ has already been formed. We effect a secondary induction over a $w = 0, 1, 2,\ldots$. Assume that a certain w has already been reached. Then $2^{v+1}w \leqq n$. We now have to mesh two intervals of the length n_w, m_w, i.e. $X^{(i'^0)}$ with $i = 2^{v+1}w+1,\ldots, 2^{v+1}w+n_w$ and $X^{(j'^0)}$ with $j + 2^{v+1}w + n_w + 1,\ldots, 2^{v+1}w+n_w+m_w$. Ordinarily $n_w = m_w = 2^v$, i.e. this is the case when both these intervals can be accomodated in the interval $1,\ldots,n$. Thus the condition for this case is $2^{v+1}(w+1) \leqq n$. Otherwise, i.e. for $2^{v+1}(w+1) > n$, we have two possibilities: First: $n_w = 2^v$ if this interval can be accomodated in the interval $1,\ldots,n$, i.e. if $2^{v+1}w+2^v \leqq n$. m_w is then chosen so as to exhaust what is left of the interval $1,\ldots,n$, i.e. $m_w = n - (2^{v+1}w + 2^v)$. Thus the condition for this case is $2^{v+1}(w+1) > n \geqq 2^{v+1}w + 2^v$. We may include $2^{v+1}(w+1) = n$, too, since the formulae of the present case coincide then with those of the previous one. Second: Otherwise, i.e. for $2^{v+1}w + 2^v > n$, the second interval is missing $m_w = 0$. n_w is then chosen so as to exhaust what is left of the interval $1,\ldots,n$, i.e. $n_w = n - 2^{v+1}w$. Thus the condition for this case is $2^{v+1}w + 2^v > n$ (and, of course, $\geqq 2^{v+1}w$). We may include $2^{v+1}w + 2^v = n$, too, since the formulae of the present case coincide then with those of the previous one. These rules can be summed up as follows:

$$(3)\qquad \begin{cases} n_w = 2^v, & m_w = 2^v \\ & \text{for } 2^{v+1}(w+1) \leqq n, \\ n_w = 2^v, & m_w = n - (2^{v+1}w+2^v) \\ & \text{for } 2^{v+1}(w+1) \geqq n \geqq 2^{v+1}w+2^v, \\ n_w = n-2^{v+1}w, & m_w = 0 \\ & \text{for } 2^{v+1}w + 2^v \geqq n \ (\geqq 2^{v+1}w). \end{cases}$$

If $2^{v+1}(w+1) < n$, then the (secondary) induction over w continues to w + 1; if $2^{v+1}(w+1) \geqq n$, then the induction over w stops. If the induction over w stops with a $w > 0$, then the (primary) induction over v continues to v + 1; if the induction over w stops with $w = 0$, then the induction over v stops. At any rate these meshings form the permutation S^{*v+1} . If the induction over v stops with a certain v, then its S^{*v+1} is the desired S^*.

There remains, finally, the necessity to specify the locations of S^{*v} and S^{*v+1}, which control the meshing processes that lead from S^{*v} to S^{*v+1}. Assume that S^{*v} occupies the interval of locations from $a^{(v)}$ to $a^{(v)}+n(p+1)-1$, and S^{*v+1} similarly the interval of locations from $a^{(v+1)}$ to $a^{(v+1)} + n(p+1)-1$. Then the meshing process which corresponds to a given w meshes the sequences that occupy the intervals from $a^{(v)}+2^{v+1}w(p+1)$ to $a^{(v)}+(2^{v+1}w+n_w)(p+1)-1$ and from $a^{(v)}+(2^{v+1}w+n_w)(p+1)$ to $a^{(v)}+(2^{v+1}w+n_w+m_w)(p+1)-1$, and places the resulting sequence into the interval from $a^{(v+1)}+2^{v+1}w(p+1)$ to $a^{(v+1)}+(2^{v+1}w+n_w+m_w)(p+1)-1$. The meshing process is that one of Problem 14.

We distinguish the constants of that problem by bars. The above specifications of the locations that are involved can then be expressed as follows:

(4)
$$\begin{cases} \bar{s}^0 = a^{(v)} + 2^{v+1}w \quad (p+1), \\ \bar{s}^1 = a^{(v)} + (2^{v+1}w + n_w)(p+1), \\ \bar{r} = a^{(v+1)} + 2^{v+1}w \quad (p+1), \\ \bar{n} = n_w, \\ \bar{m} = m_w, \\ \bar{p} = p \end{cases}$$

Clearly the interval from $a^{(v)}$ to $a^{(v)}+n(p+1)-1$ and the interval from $a^{(v+1)}$ to $a^{(v+1)}+n(p+1)-1$ must have no elements in common.

We need such an $a^{(v)}$, i.e. an interval from $a^{(v)}$ to $a^{(v)}+n(p+1)-1$, for every v for which S^{*v} is being formed. It is, however, sufficient to have two such intervals which have no elements in common, say from a to $a+n(p+1)-1$ and from b to $b+n(p+1)-1$. We can then let $a^{(v)}$ alternate between the two values a and b, i.e. we define

(5)
$$a^{(v)} \begin{cases} = a \text{ for } v \text{ even} \\ = b \text{ for } v \text{ odd} \end{cases}$$

Hence $a^{(0)} = a$, but $S^{*0} = S$, hence the statement of Problem 15 requires $a^{(0)} = s$. Consequently

(6)
$$s = a.$$

Thus a is the s of Problem 14, and b is any such location such that the interval from a to $a+n(p+1)-1$ and the interval from b to $b+n(p+1)-1$ have no elements in common.

We want the final S^* to occupy the location of the original S, i.e. we want $a^{(v+1)} = a^{(0)} = a$ for the final v with $S^{*v+1} = S^*$. I.e. this v+1 should be even. We saw further above, that the induction over v can stop when the induction over w stops with w=0, i.e. when $2^{v+1} \geqq n$. This v+1 may be odd; let us therefore agree to perform one more, seemingly superfluous, step from v to v+1. I.e. we will terminate the induction over v when

(7)
$$2^{v+1} \geqq n \text{ and } v+1 \text{ even, i.e. } a^{(v+1)} = a.$$

Hence this $v+1$ is the smallest even λ with $2^{\lambda} \geqq n$, i.e. $2\lambda'$ for the smallest integer λ' with $2^{2\lambda'} \geqq n$; i.e. $4^{\lambda'} \geqq n$. This means that

(8)
$$v+1 = 2 \langle {}^4\log n \rangle .$$

In the coded sequence that we will develop for our problem, the coded sequence that solves Problem 14 will occur as a part, as we have already pointed out. This is the coded sequence 0-43 in 11.3. We propose to use it now, and to adjust the coded sequence that we are forming accordingly.

This is analogous to what we did in 10.6 and 10.7 for the Problems 13.a and 13.b: There the coded sequence formed in 10.5 for Problem 12 was used as part of the coded sequences of the two former problems. There is, however, this difference: In 10.6, 10.7 the subsidiary sequence (of Problem 12) was attached to the end of the main sequences (of Problems 13.a, b), while now the subsidiary sequence (of Problem 14) will occur in the interior of the main sequence (of Problem 15) -- indeed it is a part of the inductive step in the double induction over v, w. The constants of Problem 14 have already been dealt with: They must be substituted according to (4) above.

In assigning letters to the various storage areas to be used, it must be remembered, just as at the corresponding points in 10.6, 10.7, that the coded sequence that we are now developing is to be used in conjunction with (i.e. as an extension of) the coded sequence of 11.3. It is therefore again necessary to classify the storage areas required by the latter: We have the storage areas C-E, which are incorporated in the final enumeration (they are 30-41 of the 0-41 of 11.1); and A^0, A^1 and B (i.e. $\bar{s}^0, \ldots, \bar{s}^0 + \bar{n}(\bar{p}+1)-1$; $\bar{s}^1, \ldots, \bar{s}^1 + \bar{m}(\bar{p}+1)-1$ and $\bar{r}, \ldots, \bar{r} + (\bar{n}+\bar{m})(\bar{p}+1)-1$), which will be part of our present A, B (i.e. of $a, \ldots, a + n(p+1)-1$ and $b, \ldots, b + n(p+1)-1$, they will be $a^{(v)} + 2^{v+1}w(p+1), \ldots, a^{(v)} + (2^{v+1}w + n_w)(p+1)-1$; $a^{(v)} + (2^{v+1}w + n_w)(p+1), \ldots, a^{(v)} + (2^{v+1}w + n_w + m_w)(p+1)-1$, and $a^{(v+1)} + 2^{v+1}w(p+1), \ldots, a^{(v+1)} + (2^{v+1}w + n_w + m_w)(p+1)-1$, cf. (4) above). Therefore we can, in assigning letters to the various storage areas in the present coding, disregard those of 11.3.

Let A and B be the storage areas corresponding to the intervals from a to $a+n(p+1)-1$ and from b to $b+n(p+1)-1$. In this way their positions will be $A.1, \ldots, n(p+1)$ and $B.1, \ldots, n(p+1)$, where A.i and B.i correspond to $a+i-1$ and $b+i-1$. A will store S at the beginning and S^* at the end of the procedure. B may be at any available place, and we assume it to be irrelevantly occupied. In the course of the procedure A and B are the alternate values of $A^{(v)}$ (for $a^{(v)} = a$ and b, i.e. for v even and odd, respectively), and $S^{(v)}$ is stored at $A^{(v)}$ while $S^{(v)}$ or $S^{(v+1)}$ is being formed. All these storages are arranged like the storage of S at A^0 in 11.3.

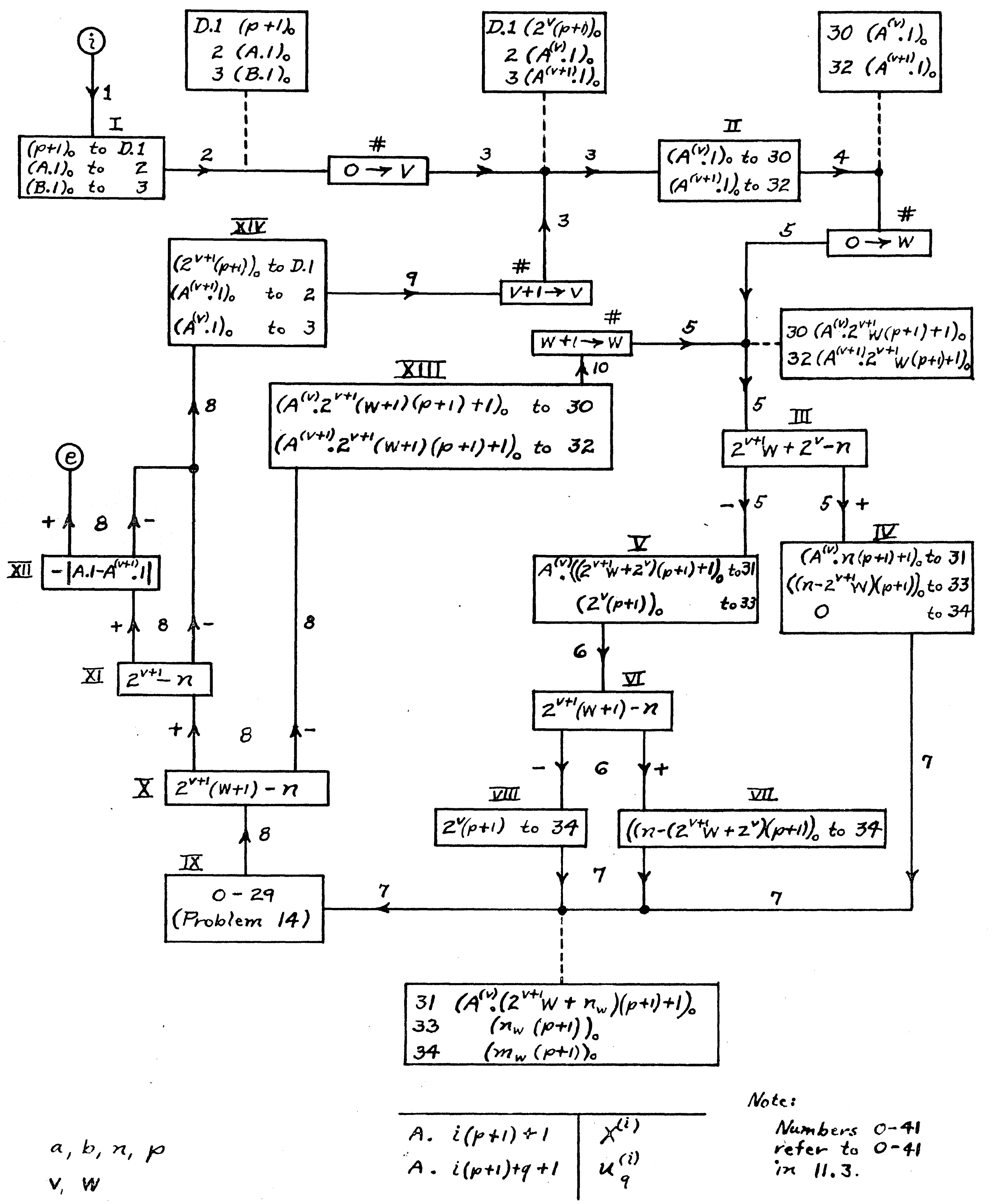
The given data of the problem, a, b, n, p, will be stored in the storage area C, with the exception of p. (It will be convenient to store them as a_0, b_0, $n(p+1)_0$. A 0 will also be stored in C.) p is also needed in the coded sequence of 11.3, and it is the only one of the constants of that problem which is independent of the induction indices v, w. (Cf. (4) above.) Hence its storage in that sequence, at 35 (as $(p+1)_0$) is adequate for our present purpose. The other constants of that problem depend on v, w (cf. above), hence their locations, 30-34, must be left empty (or, rather, irrelevantly occupied), when our present coded sequence begins to operate, and they must be appropriately substituted by its operation.

The induction index v will be stored in D. (It will be convenient to form $(2^{v+1}(p+1))_0$, and to store $(a^{(v)})_0, (a^{(v+1)})_0$ along with it.) The induction index w is part of the expressions which have to be placed at 30 and 32 ($(\bar{s}^0)_0$ and $(\bar{r})_0$, i.e. by (4) $(a^{(v)}+2^{v+1}w\,(p+1))$ and $(a^{(v+1)}+2^{v+1}w\,(p+1))_0$), hence it requires no other storage.

We can now draw the flow diagram, as shown in Figure 11.2. The actual coding obtains from this quite directly, in one instance (at the beginning of box VI) the accumulator is used as storage between two boxes.

The static coding of the boxes I-XIV follows.

C.1	a_0			
2	b_0			
I,1	35		Ac	$(p+1)_0$
2	D.1	S	D.1	$(p+1)_0$
3	C.1		Ac	a_0
4	D.2	S	D.2	a_0
5	C.2		Ac	b_0
6	D.3	S	D.3	b_0
(to II,1)				
II,1	D.2		Ac	$(a^{(v)})_0$
2	30	S	30	$(a^{(v)})_0$
3	D.3		Ac	$(a^{(v+1)})_0$
4	32	S	32	$(a^{(v+1)})_0$
(to III,1)				
III,1	30		Ac	$(a^{(v)}+2^{v+1}w\,(p+1))_0$
2	D.2	h-	Ac	$(2^{v+1}w\,(p+1))_0$
3	D.1	h	Ac	$((2^{v+1}w + 2^{v})(p+1))_0$
C.3	$(n(p+1))_0$			
III,4	C.3	h-	Ac	$((2^{v+1}w+2^{v}-n)(p+1))_0$
5	IV,1	Cc		
(to V,1)				
IV,1	C.3		Ac	$(n(p+1))_0$
2	D.2	h	Ac	$(a^{(v)}+n(p+1))_0$
3	31	S	31	$(a^{(v)}+n(p+1))_0$
4	30	h-	Ac	$((n-2^{v+1}w)(p+1))_0$
5	33	S	33	$((n-2^{v+1}w)(p+1))_0$
C.4	0			
IV,6	C.4		Ac	0
7	34	S	34	0
(to IX,1)				



D.1 (p+1)₀ 2 (A.1)₀ 3 (B.1)₀
D.1 (2^v(p+1))₀ 2 (A^(v).1)₀ 3 (A^(v+1).1)₀
30 (A^(v).1)₀ 32 (A^(v+1).1)₀
(p+1)₀ to D.1 (A.1)₀ to 2 (B.1)₀ to 3
0 → V
(A^(v).1)₀ to 30 (A^(v+1).1)₀ to 32
0 → W
(2^(v+1)(p+1))₀ to D.1 (A^(v+1).1)₀ to 2 (A^(v).1)₀ to 3
V+1 → V
W+1 → W
30 (A^(v).2^(v+1)W(p+1)+1)₀ 32 (A^(v+1).2^(v+1)W(p+1)+1)₀
(A^(v).2^(v+1)(W+1)(p+1)+1)₀ to 30 (A^(v+1).2^(v+1)(W+1)(p+1)+1)₀ to 32
2^(v+1)W + 2^v − n
−|A.1−A^(v+1).1|
A^(v).((2^(v+1)W+2^v)(p+1)+1)₀ to 31 (2^v(p+1))₀ to 33
(A^(v).n(p+1)+1)₀ to 31 ((n−2^(v+1)W)(p+1))₀ to 33 0 to 34
2^(v+1) − n
2^(v+1)(W+1) − n
2^(v+1)(W+1) − n
2^v(p+1) to 34
((n−(2^(v+1)W+2^v))(p+1))₀ to 34
0 − 29 (Problem 14)
31 (A^(v).(2^(v+1)W + n_w)(p+1)+1)₀ 33 (n_w(p+1))₀ 34 (m_w(p+1))₀
A. i(p+1)+1 | X^(i)
A. i(p+1)+q+1 | u_q^(i)
a, b, n, p
v, W
Note: Numbers 0-41 refer to 0-41 in 11.3.

FIGURE 11.2

V,1	D.1		Ac	$(2^{v}(p+1))_0$
2	33	S	33	$(2^{v}(p+1))_0$
3	30		Ac	$(a^{(v)}+2^{v+1}w\ (p+1))_0$
4	D.1	h	Ac	$(a^{(v)}+(2^{v+1}w+2^{v})(p+1))_0$
5	31	S	31	$(a^{(v)}+(2^{v+1}w+2^{v})(p+1))_0$
(to VI,1)				
VI,1	D.2	h-	Ac	$((2^{v+1}w+2^{v})(p+1))_0$
2	D.1	h	Ac	$(2^{v+1}(w+1)(p+1))_0$
3	C.3	h-	Ac	$((2^{v+1}(w+1)-n)(p+1))_0$
4	VII,1	Cc		
(to VIII,1)				
VII,1	C.3		Ac	$(n(p+1))_0$
2	30	h-	Ac	$((n-2^{v+1}w)(p+1)-a^{(v)})_0$
3	D.2	h	Ac	$((n-2^{v+1}w)(p+1))_0$
4	D.1	h-	Ac	$((n-(2^{v+1}w+2^{v}))(p+1))_0$
5	34	S	34	$((n-(2^{v+1}w+2^{v}))(p+1))_0$
(to IX,1)				
VIII,1	D.1		Ac	$(2^{v}(p+1))_0$
2	34	S	34	$(2^{v}(p+1))_0$
(to IX,1)				
IX	(Problem 14:	0-29.)		
(to X,1)				
X,1	30		Ac	$(a^{(v)}+2^{v+1}w(p+1))_0$
2	D.2	h-	Ac	$(2^{v+1}w(p+1))_0$
3	D.1	h	Ac	$((2^{v+1}w+2^{v})(p+1))_0$
4	D.1	h	Ac	$(2^{v+1}(w+1)(p+1))_0$
5	C.3	h-	Ac	$((2^{v+1}(w+1)-n)(p+1))_0$
6	XI,1	Cc		
(to XIII,1)				
XI,1	D.1		Ac	$(2^{v}(p+1))_0$
2	D.1	h	Ac	$(2^{v+1}(p+1))_0$
3	C.3	h-	Ac	$((2^{v+1}-n)(p+1))_0$
4	XII,1	Cc		
(to XIV,1)				
XII,1	C.1		Ac	a_0
2	D.3	h-	Ac	$(a-a^{(v+1)})_0$
3	s.1	S	s.1	$(a-a^{(v+1)})_0$
4	s.1	-M	Ac	$-\lvert(a-a^{(v+1)})_0\rvert$
5	e	Cc		
(to XIV,1)				

XIII,1	30		Ac	$(a^{(v)}+2^{v+1}w\ (p+1))_0$
2	D.1	h	Ac	$(a^{(v)}+(2^{v+1}w+2^{v})(p+1))_0$
3	D.1	h	Ac	$(a^{(v)}+2^{v+1}(w+1)(p+1))_0$
4	30	S	30	$(a^{(v)}+2^{v+1}(w+1)(p+1))_0$
5	32		Ac	$(a^{(v+1)}+2^{v+1}w\ (p+1))_0$
6	D.1	h	Ac	$(a^{(v+1)}+(2^{v+1}w+2^{v})(p+1))_0$
7	D.1	h	Ac	$(a^{(v+1)}+2^{v+1}(w+1)(p+1))_0$
8	32	S	32	$(a^{(v+1)}+2^{v+1}(w+1)(p+1))_0$
	(to III,1)			
XIV,1	D.1		Ac	$(2^{v}(p+1))_0$
2	D.1	h	Ac	$(2^{v+1}(p+1))_0$
3	D.1	S	Ac	$(2^{v+1}(p+1))_0$
4	D.2		Ac	$(a^{(v)})_0$
5	s.1	S	s.1	$(a^{(v)})_0$
6	D.3		Ac	$(a^{(v+1)})_0$
7	D.2	S	D.2	$(a^{(v+1)})_0$
8	s.1		Ac	$(a^{(v)})_0$
9	D.3	S	D.3	$(a^{(v)})_0$
	(to II,1)			

The ordering of the boxes is I, II, III, V, VI, VIII, IX, X, XIII; XI, XIV; IV; VII; XII, and IX, IX, XIV, III, II must also be the immediate successors of IV, VII, XII, XIII, XIV, respectively. In addition, IX cannot be placed immediately after VIII, since IX,1 is 0 -- but IX must nevertheless be the immediate successor of VIII. All this necessitates the extra orders

IV,8	IX,1	C
VII,6	IX,1	C
VIII,3	IX,1	C
XII,6	XIV,1	C
XIII,9	III,1	C
XIV,10	II,1	C

Finally in order that X be the immediate successor of IX, the e of 0-41 (in 26) must be equal to X, 1.

We must now assign C.1-4, D.1-3, s.1 their actual values, pair the 76 orders I,1-6, II,1-4, III,1-5, IV,1-8, V,1-5, VI,1-4, VII,1-6, VIII,1-3, X,1-6, XI,1-4, XII,1-6, XIII,1-9, XIV,1-10 to 38 words, and then assign I,1 - XIV,10 their actual values. (IX is omitted, since it is contained in 0-41.) We wish to do this as a continuation of the code of 11.3. We will therefore begin with the number 42. Furthermore the contents of D.1-3, s.1 are irrelevant, like those of 37-41 there. We saw, in addition, that 30-35 (with the exceptions of 30,32) are also irrelevant. We must, however, make this reservation: 30-41 are needed while box IX operates, and during this period D.1-3 are relevantly occupied, but not s.1. s.1 is relevantly occupied only while boxes XII, XIV operate, and during this period 30-35 and 37-41 are irrelevant. Hence only s.1 can be made to coincide with one of these, say with 41. Summing all these things up, we obtain the following table:

I,1-6	42 -44'	VIII,1-3	54 -55	IV,1-8	70 -73'
II,1-4	45 -46'	X,1-6	55'-58	VII,1-6	74 -76'
III,1-5	47 -49	XIII,1-9	58'-62'	XII,1-6	77 -79'
V,1-5	49'-51'	XI,1-4	63 -64'	C.1-4	80 -83
VI,1-4	52 -53'	XIV,1-10	65 -69'	D.1-3	84 -86
				s.1	41

Now we obtain this coded sequence:

42	35 ,	84 S	57	84 h ,	82 h-	72	33 S ,	83
43	80 ,	85 S	58	63 Cc,	30	73	34 S ,	0 C
44	81 ,	86 S	59	84 h ,	84 h	74	82 ,	30 h-
45	85 ,	30 S	60	30 S ,	32	75	85 h ,	84 h-
46	86 ,	32 S	61	84 h ,	84 h	76	34 S ,	0 C
47	30 ,	85 h-	62	32 S ,	47 C	77	80 ,	86 h-
48	84 h ,	82 h-	63	84 ,	84 h	78	41 S ,	41 -M·
49	70 Cc,	84	64	82 h-,	77 Cc	79	e Cc,	65 C
50	33 S ,	30	65	84 ,	84 h	80	a_0	
51	84 h ,	31 S	66	84 S ,	85	81	b_0	
52	85 h-,	84 h	67	41 S ,	86	82	$(n(p+1))_0$	
53	82 h-,	74 Cc	68	85 S ,	41	83	0	
54	84 ,	34 S	69	86 S ,	45 C	84	-	
55	0 C ,	30	70	82 ,	85 h	85	-	
56	85 h-,	84 h	71	31 S ,	30 h-	86	-	

For the sake of completeness, we restate that part of 0-41 of 11.3, which contains all changes and all substitutable constants of the problem. This is 26 and 30-41:

26	55 Cc', 10 Cc'	33	-	37	-
30	-	34	-	38	-
31	-	35	-	39	-
32	-	36	1_0	40	-
				41	-

The durations may be estimated as follows:

I: 225 μ, II: 150 μ, III: 200 μ, IV: 300 μ, V: 200 μ, VI: 150 μ, VII: 225 μ, VIII: 125 μ, X: 225 μ, XI: 150 μ, XII: 225 μ, XIII: 350 μ, XIV: 375 μ.

IX: The precise estimate at the end of 11.3 is

$$((575\, p + 1,125)\,(n_w + m_w) + 575)\ \mu.$$

Total: Put $\bar{v} = 2 \langle {}^4\log n \rangle$, $\bar{\eta} = 2 \langle {}^4\log n \rangle - \langle {}^2\log n \rangle = 0$ or 1, $\bar{w} = \left\langle \frac{n}{2^{v+1}} \right\rangle$. Then the total is

$$I + \sum_{v=0}^{\bar{v}-1} \left\{ II + \sum_{w=0}^{\bar{w}-1} \left[\begin{array}{c} III + (IV^{*)} \text{ or } (V + VI + (VII^{*)} \text{ or } VIII))) \\ + IX + X + (XI^{*)} \text{ or } XIII) \end{array} \right] + XIV^{**)} \right\} + XII\ (\bar{\eta}+1)$$

This is majorized by

$$(225 + \sum_{v=0}^{\bar{v}-1} \left\{ 150 + \sum_{w=0}^{\bar{w}-1} \left[\begin{array}{c} 200 + 200 + 150 + 125 + (575\ p + 1{,}125) \times \\ \times (n_w + m_w) + 575 + 225 + 350 \end{array} \right] + 100 - 200 + 375 \right\} -$$

$$- 375 + 225\ (\bar{\eta}+1))\ \mu\ .$$

Since $\sum_{w=0}^{\bar{w}-1} (n_w + m_w) = n$, $\bar{\eta} \leqq 1$, this is majorized by

$$(225 + \sum_{v=0}^{\bar{v}-1} \left\{ (575\ p + 1{,}125)\ n + 1{,}825 \left\langle \frac{n}{2^{v+1}} \right\rangle + 425 \right\} + 75)\ \mu\ .$$

Since $\sum_{v=0}^{\bar{v}-1} \left\langle \frac{n}{2^{v+1}} \right\rangle \leqq \sum_{v=0}^{\bar{v}-1} \left(\frac{n}{2^{v+1}} + 1 \right) \leqq n + \bar{v}$, this is majorized by

$$((575\ p + 1{,}125)\ n\ \bar{v} + 1{,}825\ n + 2{,}250\ v + 300)\ \mu\ .$$

We can write this in this form:

$$((575\ p + 1{,}200\ (1 + \delta))\ n\ \bar{v}\ )\ \mu \approx (.6\ (p + 2\ \ (1 + \delta))\ n\ \bar{v})\ m\ ,$$

where

$$\bar{v} = 2 \langle {}^4\log n \rangle\ ,$$

$$\delta = -\ .06 + \frac{1.52}{\bar{v}} + \frac{1.83}{n} + \frac{.25}{n\ \bar{v}}\ .$$

δ is a small and slowly changing quantity: For n = 100; 1,000; 10,000 (the last value is, of course, incompatible with the memory capacities which appear to be practical in the near future) we have $\bar{v}$ = 8; 10; 14 and hence δ = .15; .09; .05, respectively.

---

*) At most once among the w in $\sum_{w=0}^{\bar{w}-1}$. We disregard this effect for IV, which represents the shorter alternative. We replace VII, XI by their alternatives, VIII, XIII, respectively, inside the $\sum_{w=0}^{\bar{w}-1}$. This necessitates the corrections VII-VIII = 100 μ, XI-XIII = -200 μ, respectively, outside the $\sum_{w=0}^{\bar{w}-1}$. (Cf. the third and second terms from the right in the brackets { ... } of the next formula.)

**) Missing once among the v in $\sum_{v=0}^{\bar{v}-1}$. We may therefore subtract XIV = 375 μ as a correction outside the $\sum_{v=0}^{\bar{v}-1}$.

11.5 The meshing and sorting speeds of 11.3 and 11.4 are best expressed in terms of m per complex or of complexes per minute. The number of complexes in these two problems is n + m and n, respectively, hence the number of m per complex is $\approx .6\ (p+2)$ and $\approx .6\ (p+2(1+\delta)) \cdot 2\ \langle {}^4\log n\rangle \approx 1.2\ (p+2.3)\ \langle {}^4\log n\rangle$, respectively. The number of complexes per minute obtains by dividing these numbers into 60,000, i.e. it is

(9) $$\frac{100,000}{p+2} \quad \text{and} \quad \frac{50,000}{(p+2.3)\langle {}^4\log n\rangle}, \text{ respectively.}$$

To get some frame of reference in which to evaluate these figures, one might compare them with the corresponding speeds on standard, electro-mechanical punch-card equipment.

Meshing can be effected with the I.B.M. collator, which has a speed of 225 cards per minute. Sorting can be effected by repeated runs through the I.B.M. (decimal) sorter. The sorting method which is then used is not based on iterated meshing, and hence differs essentially from our method in 11.4. Strictly construed, it requires as many runs through the sorter, as there are decimal digits in the principal number. This number of digits is usually between 3 and 8. Since we are dealing in our case with 40 binary digits, corresponding to 12 decimal digits, the use of the value 6 in such a comparison does not seem unfair. The sorter has a normal speed of 400 cards per minute, and it can unquestionably be accelerated beyond this level. 50% seems to be a reasonable estimate of this potential acceleration. Making the comparison on this basis, we have a speed of $\frac{400 \times 1.5}{6} = 100$ cards per minute. Hence we have these speeds for meshing and sorting in cards per minute:

(10) 225 and 100, respectively.

A card corresponds to one of our complexes, since it moves as a unit in meshing and in sorting. Hence the speeds of (9) and (10) are directly comparable, and they give these ratios:

(11) $$\frac{(9)}{(10)} \approx \frac{444}{p+2} \quad \text{and} \quad \frac{500}{(p+2.3)\langle {}^4\log n\rangle}, \text{ respectively.}$$

A standard I.B.M. punch card has room for 80 decimal digits. This is equivalent to about 270 binary digits, i.e. somewhat less than 7 of our 40 binary digit numbers. It is, of course, in most cases not used to full capacity. Hence it is best compared to a complex with $p+1 \leqq 7$, i.e. $p = 1, \ldots, 6$. For n values from 100 to 1,000 seem realistic (in view of the probably available "inner" memory capacities, and assuming that no "outer" memory is used, cf. the second remark at the end of this section), hence $\langle {}^4\log n\rangle = 4, 5$. Consequently the ratios of (11) become

(11') $$\frac{(9)}{(10)} \approx 150 \text{ to } 55 \text{ and } 30 \text{ to } 15, \text{ respectively.}$$

In considering these figures the following facts should be kept in mind:

First: In using an electronic machine of the type under consideration for sorting problems, one is using it in a way which is least favorable for its specific characteristics, and most favorable for those of a mechanical punch card machine. Indeed, the coherence of a complex $X = (x; u_1, \ldots, u_p)$, i.e. the close connection between the movements of its principal number x and the subsidiary numbers $u_1, \ldots, u_p$, is guaranteed in a punch card machine by the physical identity and coherence of the punch card which it occupies, while in the electronic machine in question x and each $u_1, \ldots, u_p$ must be transferred separately; there is no intrinsic coherence between these items, and their ultimate, effective coherence results by synthesis from a multitude of appropriately coordinated individual transfers. This situation is very different from the one which exists in properly mathematical problems, where multiplication is a decisive operation, with the result that the extreme speed of the basic electronic organs can become directly effective, since the electronic multiplier is just as efficiently organized and highly paralleled as its mechanical and electromechanical counterparts. Thus in properly mathematical problems the ratio of speed is of the order of the ratio of the, say, .1 m multiplication time of an electronic multiplier and of the, say, 1 to 10 second multiplication time of an electromechanical multiplier, i.e., say, 10,000 to 100,000 -- while (11') above gave speed ratios of only 15 to 150.

Second: The "inner" memory capacities of the electronic machines that we envisage will hardly allow of values of n (and m) in excess of about 1,000. When this limit is exceeded, i.e. when really large scale sorting problems arise, then the "outer" memory (magnetic wire or tape, or the like) must also be used. The "inner" memory will then handle the problem in segments of several 100, or possibly up to 1,000, complexes each, and these are combined by iterated passages to and from the "outer" memory. This requires some additional coded instructions, to control the transfers to and from the "outer" memory, and slows the entire procedure somewhat, since the "outer" memory is very considerably less flexible and less fast available than the "inner" one. Nevertheless, this slowdown is not very bad: We saw in 11.3 that meshing requires .6 m per number. Magnetic wires or tapes can certainly be run at speeds of 20,000-40,000 (binary) digits per second, i.e. 500-1,000 (40 binary digit words or) numbers per second. This means 1-2 m per number. Thus the times required for each one of these two phases of the matter are of the same order of magnitude. In addition to this the "outer" memory is likely to be of a multiple, parallel channel type.

We will discuss this large scale sorting problem later, when we come to the use of the "outer" memory. It is clear, however, that it will render the comparison of speeds, (11'), somewhat less favorable.

Third: We have so far emphasized the unfavorable aspects of sorting with an electronic machine of the type under consideration -- i.e. one which is primarily an all-purpose, mathematical machine, not conceived particularly for sorting problems. Let us now point out the favorable aspects of the matter. These seem to be the main ones:

(a) All the disadvantages that we mentioned are relative ones (i.e. in comparison to the properly mathematical problems) -- electronic sorting should nevertheless be faster than mechanical, punch card sorting: In our above examples by factors of the order of 10 to 100.

(b) The results of electronic sorting appear in a form which is not exposed to the risks of the unavoidable human manipulations of punch cards: They are in the "inner" memory of the machine, which requires no human manipulation whatever; or in the "outer" memory, which is a connected, physically stable medium like magnetic wire or tape.

(c) The sorting operations can be combined with and alternated with properly mathematical operations. This can be done entirely by coded instructions under the inner control of the machine's own control organs, with no need for human intervention and no interruption of the fully automatic operation of the machine. This circumstance is likely to be of great importance in statistical problems. It represents a fundamental departure from the characteristics of existing sorting devices, which are very limited in their properly mathematical capabilities.