

Regnecentralen
Gl. Carlsbergvej 2
Valby
10.11.1960.

SORTERING OG ANDEN DATABEHANDLING

Jørn Jensen

Indhold.

Indledning	2
Betegnelser og begreber	2
DA 1 Simpel sortering i ferritlageret	4
DA 2 Hurtig sortering i ferritlageret	5
DA 20 Trim DA 2	5
DA 3 Sortering af data paa tromle	6
DA 30 Trim DA 3	6
DA 4a Sortering af data paa baand	7
DA 40a Trim DA 40	8
DA 5 Checksum og kopiering	9
DA 7 Datafunktion	10
DA 70 Trim DA 7	12

Bemærk: Programmet DA 4 erstattes af DA 4a og er derfor ikke medtaget.

INDLEDNING

Ved kørsel af databehandlingsopgaver skal der udføres ret simple operationer paa et stort antal ens opbyggede datasæt.

Et typisk forløb er for eksempel dette:

1. Indlæsning af en række posteringer (bevægelser paa konti) og lagring af det indlæste (normalt paa magnetbaand).
2. Sortering af det indlæste efter kontonummer.
3. Dannelse af nye saldi udfra de indlæste, sorterede posteringer og gamle saldi (normalt opbevaret paa magnetbaand).
4. Indsortering af de indlæste, sorterede posteringer i tidligere behandlede posteringer.
5. Opsøgning (for dannelse af kontoudtog eller lignende) af visse konti eller grupper af konti.

Første del af den ovenfor skitserede proces (indlæsning) er som regel stærkt opgaveafhængig og maa derfor kodes til hver enkelt opgave, hvorimod de næste delprocesser (sortering, dannelse af nye saldi, sammenfletning med ældre posteringer) alle er ret ensartede og derfor med fordel vil kunne gøre brug af dataadministrationssekvenserne, i det følgende kaldt DA-sekvenserne.

Nedenfor gives en oversigt over disse DA-sekvenser.

I denne oversigt er der anvendt ensartede betegnelser for de begreber, der opereres med, det er derfor nødvendigt at indlede med en omtale af disse.

BETEGNELSER OG BEGREBER

1. Anvendte betegnelser.

Datamængde	M	eventuelt	M1, M2, . . .
Antal datasæt i M	N	-	N1, N2, . . .
Antal datasæt pr blok	n	-	n1, n2, . . .
i'te datasæt	S[i]	-	S1[i], S2[i], . . .
i'te datasæts hoved	H[i]	-	H1[i], H2[i], . . .
Antal heo pr datasæt	l	-	l1, l2, . . .
Baand, baandstationsnr.	B	-	B1, B2, . . .
Begyndelsesbloknr. for M	b	-	b1, b2, . . .
Begyndelsesadresse i FL	L	-	L1, L2, . . .

1.1. Datamængde.

DA-sekvenserne opererer paa datamængder, der som oftest forudsættes lagret paa magnetbaand.

En datamængde M bestaar af N lige lange datasæt.

Hvert datasæt bestaar af l heo, af hvilke den identificerende del (kontonummer eller lignende) kaldes datasættets hoved H. Hvis intet andet er angivet, forudsættes H at være første heo i datasættet.

1.2. Lagring af datamængde paa magnetbaand.

En datamængde M lagret paa magnetbaand forudsættes lagret i konsekutive blokke, begyndende paa blok b paa baandstation B.

(B er ydre-enhedsnummeret for vedkommende baandstation (2001, 2002 eller 2003)).

Paa hver blok er lagret n datasæt ($n \times l \leq 128$), saaledes at første datasæt paa blokken begynder i heo 0, næste i heo $2 \times l$ osv.

Sidste blok i datamængden kan dog indeholde færre end n datasæt.

NB: de enkelte blokke behøver altsaa ikke være fyldt helt op, men indholdet af eventuelle overskydende heo paa blokkene vil ikke være defineret, dersom datamængden er blevet behandlet af en DA-sekvens.

2. DA-sekvenser og deres trimming.

DA-sekvenserne benævnes DA1, DA2, . . .

Adskillige af DA-sekvenserne kræver at være trimmet før anvendelsen. Den til DA2 hørende trimmesekvens benævnes DA20, den til DA3 hørende benævnes DA30 osv.

For samtlige DA-sekvenser, der kræver trimming, gælder det, at denne kun behøver at udføres een gang, dersom trimmeparametrene er uændrede fra gang til gang DA-sekvensen benyttes. Trimmingen kan da (for anvendelse til et givet program) udføres ved en særskilt kørsel og den trimmede DA-sekvens kan udskrives. Dette sparer plads ved kørslen.

3. Sortering.

Sortering af en datamængde består i en flytning af datasættene, der indgaar, saaledes at dersom $H[i] < H[j]$ placeres $S[i]$ før $S[j]$.

H forudsættes at være første heo i datasættet. For mulige indgreb for sortering paa et andet, eller længere, hoved, henvises til de detaljerede beskrivelser af de enkelte sekvenser.

NB: Dersom det kun er den forreste del af første heo, der udgør hovedet, foregaar sorteringen, som defineret ovenfor, selvfølgelig korrekt.

Det kræves iøvrigt at $-1 \leq H < 1 - 2^{-39}$.

OVERSIGT OVER DA-SEKVENSERNE.

Den følgende oversigt omfatter kun de nødvendige oplysninger for brugen af DA-sekvenserne, medens de egentlige kodebeskrivelser udgives senere.

BA1 SIMPEL SORTERING I FERRITLAGERET

Kodelængde: 0 - 63
Begyndelsesadresse: lige.

Indgang: N datasæt à 1 heo lagret fra og med hec L i ferritlageret.

Indhop: O A8 16
 N A 00
 2x1 A 00
 L A 00

Udgang: N datasæt sorteret og lagret fra og med hec L i ferritlageret,
(Udhop 4D10).

Køretid: Ca. $15 \times N^{3/2}$ AT.

DA2 HURTIG SORTERING I FERRITLAGERET

Kodelængde: 0 - 119.
Begyndelsesadresse: lige.
Arbejdsceller: $2 \times 1 \times N$ hvo
fra og med hec L2.

Indgang: N datasæt a l heo lagret fra og med hec L1 i ferritlageret.

DA2 trimmet med DA20; trimning kan dog undværes dersom l
er uændret siden sidste trimning.

Indhop: O A8 16
 N A 00
 L1 A 00
 L2 A 00

Udgang: N datasæt sorteret og lagret fra og med hec L i ferritlageret.
C(ARvadr) = L.

Hvis $2^{2 \times p} \leq N < 2^{2 \times p + 1}$, L = L1,
ellers L = L2.
(Udhop 4D10).

Køretid: Ca $N \times \log_2 N \times (10 + 2 \times 1)$ AT.

DA20 TRIM DA2

Kodelængde: 0 - 57.
Begyndelsesadresse: lige.

Indgang: DA2 lagret fra OA9 og fremefter.

Indhop: O A8 16
 2x1 A 00

Udgang: DA2 trimmet til sortering af datasæt a l heo.
NB. Trimning kan udelades dersom l er uændret siden sidste
trimning.
(Udhop 2D10).

Køretid: Ca 70 AT.

DA3 SORTERING AF DATA PAA TROMLE

Kodelængde: 0 - 279.
Begyndelsesadresse: lige.
Arbejdsceller: a hao i for-
længelse af DA3, se nedenfor.

Indgang: N datasæt lagret fra og med heo 0 paa kanal k med n datasæt
à 1 heo pr 'blok'. En 'blok' er defineret saaledes at
hvert n'te datasæt skal begynde i heo 0 paa en kanal.
De n datasæt skal staa tæt efter hinanden og maa højst fylde
3 kanaler, d.v.s. $n \times l \leq 192$.
(Dersom $n \times l > 64$ kræves mange arbejdsceller).

DA3 trimmet med DA30, trimming kan dog undværes hvis $n = 32$
og $l = 1$ og/eller n og l er uændrede fra sidste trimming.

a hao i forlængelse af DA3 ledige.
 $a = 192 \times \text{antal kanaler pr 'blok'} + \text{samlet antal kanaler}$.

Indhop: O A8 16
C N F (N $\times 2^{-19}$, $n < N$)
k A 00

Udgang: N datasæt sorteret og lagret fra og med heo 0 paa kanal k
med n datasæt pr 'blok'.
Er N ikke delelig med n vil sidste 'blok' være fyldt op med
datasæt med $H = 1 - 2^{-39}$.

Indholdet af eventuelle rester af kanaler mellem 2 paa
hinanden følgende 'blokke' er undefineret.

(Udhop 3D10).

DA30 TRIM DA3

Kodelængde: 0 - 67
Begyndelsesadresse: lige.

Indgang: DA3 lagret fra OA9 og fremefter.

Indhop: O A8 16
n A 00
2x1 A 00

Udgang: DA3 trimmet til sortering af datamængder lagret med n datasæt
à 1 heo pr 'blok'.
(Udhop 3D10).

Køretid: Ca 80 AT.

DA4a SORTERING AF DATA PAA BAAND

Kodelængde: 0 - 445.
Begyndelsesadresse: lige.
Arbejdsceller: 390 hao
i forlængelse af DA4a.

Indgang: N datasæt a l heo lagret paa baand B1 fra og med blok b1 med n datasæt pr blok.

BA2 intakt og alle 3 baand startet med BA1.

DA4a trimmet af DA40a, trimming kan dog undværes dersom samtlige trimmeparametre er uændrede siden sidste trimming.

C(AR) = helordsvis checksum af alle N datasæt.

Indhop: 0 A8 16
C N F (N × 2⁻¹⁹)

Udgang: Hvis checksum ikke stemmer ved første læsning af B1 faas straks udhop (2D10, IRB og IRC ikke retableret) med C(AR) < 0.
Hvis checksum stemmer (kontrolleres kun ved første læsning) faas udhop (2D10) med N datasæt sorteret og lagret paa baand B1 fra og med blok b1 med n datasæt pr blok. C(AR) ≥ 0.
Hvis N ikke er delelig med n, vil sidste blok være fyldt op med datasæt med H = 1 - 2⁻¹⁹.
Indholdet af eventuelle ubenyttede heo af hver blok (efter de n datasæt) er undefineret.

Fejludskr: I BA2. Desuden kontrolleres hver gang baandene skal spoles tilbage, at sidst benyttede bloknr. stemmer.
Ved fejl faas KP-udskrift fra 167A8 af:
KR 'B'baandstation korrekt bloknr. forkert bloknr.
udskriften afsluttes med '-' dersom tobitsfejl.
Derefter repeterer DA4a sidst udførte del af sorteringen og fortsætter.
Ved stadig fejl: fornyet fejludskrift, repetition osv.
Ingen stop eller tælling ved fejl.

Køretid: Ca 2 × G × (log₂ G + 6) sekunder. G = N × 1 / 1000, altsaa datamængdens samlede længde i 1000 heo.

DA40a TRIM DA4a

Kodelængde: 0 - 159.
Begyndelsesadresse: lige.

Indgang: DA4a lagret fra OA9 og fremefter.

Indhop: O A8 16
 n A 00
 2x1 A 00
 B1 A 1C
 C b1 F (b1 x 2⁻¹⁹)
 B2 A 1C
 C b2 F (b2 x 2⁻¹⁹)
 B3 A 1C
 C b3 F (b3 x 2⁻¹⁹)

Udgang: DA4a trimmet til sortering af en datamængde lagret paa baand B1
begyndende blok b1 med n datasæt a l heo pr blok.
Hjælpebaand: B2 fra blok b2
 B3 fra blok b3.
Hjælpebaandene skal hver kunne rumme den fulde datamængde.
(Udhop 9D10).

Køretid: Ca 200 AT.

DA5 CHECKSUM OG KOPIERING

Kodelængde: 0 - 59.
Begyndelsesadresse: lige.
Arbejdsceller: 132 haa
i forlængelse af DA5.

Indgang: N datasæt a l heo lagret paa baand B1 fra og med blok b1
med n datasæt pr blok.

BA2 intakt og de aktuelle baand startet med BA1.

Indhop 1: Checksum 0 A8 16
 C N F (N x 2⁻¹⁹)
 n A 00
 2x1 A 00
 B1 A 1C
 C b1 F (b1 x 2⁻¹⁹)

Udgang 1: AR := helordsvis sum af N datasæt.
(Udhop 6D10).

Indhop 2: Kopiering 1 A8 16
 C N F (N x 2⁻¹⁹)
 n A 00
 2x1 A 00
 B1 A 1C
 C b1 F (b1 x 2⁻¹⁹)
 B2 A 1C
 C b2 F (b2 x 2⁻¹⁹)

Udgang 2: AR := helordsvis sum af N datasæt (som læst fra B1).
De N datasæt kopieret til baand B2, lagret fra og med
blok b2 med n datasæt pr blok.
(Udhop: 8D10).

Fejlstop: Kun i BA2.

DA7 DATAFUNKTION

Kodelængde: 0 - 139
Arbejdsceller: 390 hao
i forlængelse af DA7.
Begyndelsesadresse: lige.

1. Funktion.

Udfra to givne, sorterede datamængder: M1 (N1, n1, l1, B1, b1)
og: M2 (N2, n2, l2, B2, b2)
dannes en tredje: M3 (N3, n3, l3, B3, b3)

1.1 Datamængden M3.

For M3 gælder: $n_3 = n_1$, $l_3 = l_1$ og $N_3 \geq N_1$.
M3 dannes ved, at alle datasæt fra M1 overføres til M3, idet der umiddelbart efter overførslen ($S_3[k] := S_1[i]$) sker en modifikation af $S_3[k]$ med alle de datasæt $S_2[j]$, for hvilke det gælder at $H_3[k] = H_2[j]$.
Modifikationen sker ved hjælp af en udefra givet funktion $F_1(S_2[j], S_3[k])$.
Funktionen F_1 kan f.eks. være dannelsen af en ny saldo (S_3) udfra den gamle saldo (S_1), der er blevet overført til M3, og en postering (S_2), idet DA7 vil hoppe til F_1 hvergang der træffes en postering i M2 med samme kontonummer (H_2) som den til M3 overførte gamle saldo.

Træffer DA7 under sammenligningen af M1 og M2 et datasæt $S_2[j]$, der ikke svarer til et datasæt $S_1[i]$, d.v.s. $H_2[j] < H_1[i]$, vil DA7 hoppe til en anden udefra givet funktion $F_2(S_2[j], S_3[k])$. I denne funktion kan der da foretages fejludskrift (hvis der f.eks. ikke maa forekomme kontonumre i M2, der ikke forekommer i M1), eller der kan dannes et datasæt $S_3[k]$ (f.eks. oprettelse af en ny konto). Dette sidste vil bevirke at det dannede datasæt $S_3[k]$ vil blive behandlet af DA7 nøjagtig som om det stammede fra datamængden M1.

1.2 Konventioner for F_1 og F_2 .

Hop fra DA7 til F_1 eller F_2 sker med 16-hop til adresserne F_1 og F_2 , der er givet som programparametre ved trimningen af DA7 (med DA70).

Adressen paa første heo af $S_3[k]$ staar i IRB,
adressen paa første heo af $S_2[j]$ staar i IRC.

Efter udførelsen af F_1 eller F_2 sker tilbagehop saaledes:

IRB, IRC og IRD skal være retablerede,
C(AR) irrelevant.

Udhopsordre: Fra F_1 : 1D10.

Fra F_2 : Dersom der er dannet datasæt $S_3[k]$: 1D10,
ellers: 2D10.

1.3 Konventioner for M1 og M2.

M1 og M2 maa gerne have $n_1 \neq n_2$ og/eller $l_1 \neq l_2$ (der vil ofte være data i en postering, der ikke skal medtages i saldoen).

Mængden M1 kan være tom ($N_1 = 0$).

Mængden M2 skal indeholde mindst et datasæt ($N_2 \neq 0$).

1.3.1 Hovedet paa de indgaaende datasæt.

Da man ofte opererer med hoveder, der er mindre end et heo, og det i DA7 drejer sig om at konstatere identitet mellem hovederne i M1 og M2 (dette er uden betydning under sortering), er der ved anvendelsen af DA7 mulighed for at betragte de første h bits af heo 0 i et datasæt som hovede, idet h opgives som programparameter til trimmesekvensen DA70.

1.4 Kontrolsummer og optælling.

DA7 danner 3 kontrolsummer:

Sum1 = helordsvis sum af alle N1 datasæt + Sum1
Sum2 = - - - N2 - + Sum2
Sum3 = - - - N3 - + Sum3

Disse summer dannes under selve anvendelsen af datasættene i DA7, og ikke ved et specielt gennemløb.

Desuden optæller DA7 antal datasæt N3 i datamængden M3.

Adresserne paa heo Sum1, heo Sum2, heo Sum3, og hao N3 opgives som programparametre til trimmesekvensen DA70.

NB. Sum1, Sum2, Sum3, og N3 bliver ikke nulstillet af DA7 før brugen.

2. Indgang: N1 sorterede datasæt lagret paa baand B1 fra og med blok b1 med n1 datasæt á 11 heo pr blok.
N2 sorterede datasæt lagret paa baand B2 fra og med blok b2 med n2 datasæt á 12 heo pr blok.

BA2 intakt og alle 3 baand startet med BA1

DA7 trimmet med DA70, trimming kan dog undværes, dersom trimmeparametrene er uændrede siden sidste trimming.

3. Indhop: O A8 16
C N1 F $(N1 \times 2^{-19})$
C N2 F $(N2 \times 2^{-19})$

4. Udgang: N3 sorterede datasæt lagret paa baand B3 fra og med blok b3 med n1 datasæt á 11 heo pr blok.
De N3 datasæt er dannet som beskrevet under 1 (funktion).
Kontrolsummer og optælling: Se 1.4.

5. Fejlstop: Kun i BA2.

6. Køretid: Ca baandtiden for gennemløb af N1 + N2 datasæt + tid for F1 og F2. Dersom l er lille kan tiden dog blive noget større.

DA70 TRIM DA7

Kodelængde: 0 - 51
Begyndelsesadresse: lige.

Indgang: DA7 lagret fra OA9 og fremefter.

Indhop: O A8 16
 n1 A 00
 2x11 A 00
 n2 A 00
 2x12 A 00
 F1 A 00 startadresse for funktionen F1
 F2 A 00 - - - F2
 h A 00 antal bits i datasæthovedet
 B1 A 1C
 C b1 F (b1 x 2⁻¹⁹)
 B2 A 1C
 C b2 F (b2 x 2⁻¹⁹)
 B3 A 1C
 C b3 F (b3 x 2⁻¹⁹)
 L(Sum1) A 00
 L(Sum2) A 00
 L(Sum3) A 00
 L(N3) A 00

Udgang: DA7 trimmet (se DA7).
(Udhop 18D10).
NB. Trimming kan udelades dersom trimmeparametrene er
uændrede fra sidste trimming.

Køretid: 52 AT.