

# software MBASIC



## Gestione di alberi binari di Vincenzo Bossi - Torino

Dal momento che ci fa sempre piacere ricevere programmi da parte dei lettori, ecco che anche in questa puntata (con il seguito nella prossima) pubblichiamo ben volentieri un programma, ben documentato, di un nostro lettore di Torino, il quale ci ha inviato uno studio sugli alberi binari sotto forma di una subroutine che implementa tutte le principali funzioni «primitive» di gestione di tali strutture dati.

In questo numero, per motivi di spazio, parleremo soltanto dell'insieme di subroutine, mentre nel prossimo numero analizzeremo anche un esempio di programma applicativo, seguendo quanto ci dice l'autore.  
Lasciamogli dunque la parola.

### Introduzione

Il programma, che si chiama «alber0», è concepito come una subroutine, da inserire in fondo a qualsiasi programma applicativo, e da usare mediante chiamate con passaggio dei parametri necessari.

Detto programma costruisce un albero binario che conserva le «chiavi» accoppiate ai dati: questo albero binario costituisce il «FILE INDICE» (nel seguito abbreviato con «FI»), in cui oltre alle chiavi esistono tutti i puntatori necessari (ai figli destro e sinistro ed al padre) ed inoltre il puntatore al record dati, contenuto in un altro file,

il «FILE DATI», abbreviato nel seguito con «FD».

La chiave può avere una lunghezza a piacere, mentre il numero di FI attualmente gestibile è pari a 5, con altrettanti FD, tutti apribili contemporaneamente.

Se però qualche megalomane desiderasse gestire ancora più dati, lo potrà fare aggiungendo alcune linee al programma, come specificato nel paragrafo «Avvertenze».

Tutte le variabili usate da «alber0» iniziano per «H» e perciò è meglio evitare di usare nel programma chiamante delle variabili iniziati anche loro per «H»: questo perché fin dove è possibile le variabili in questione sono definite implicitamente intere.

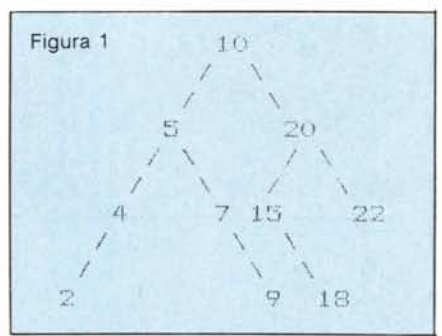
Questo fatto comporta che la dimensione del file risulta limitata ad un po' più di 35000 record: se sono necessari più record basta eliminare l'istruzione «DEFINT H» dalla linea 60120.

Il programma è stato realizzato su di un Epson QX-10, con un «Multifonts Basic vers. 1.3», un Basic abbastanza standard, se si eccettuano alcune interessanti caratteristiche grafiche quali i differenti set di caratteri, sia su stampante che su video.

Ad ogni modo «alber0» risulta assolutamente standard, mentre qualche minima modifica verrà richiesta dal «alber1», il programma di gestione che vedremo la prossima puntata.

### Teoria generale

Il programma alber0 gestisce una struttura ad alberi binari. Un albero binario è un albero in cui ogni nodo ha, al massimo, 2 figli.

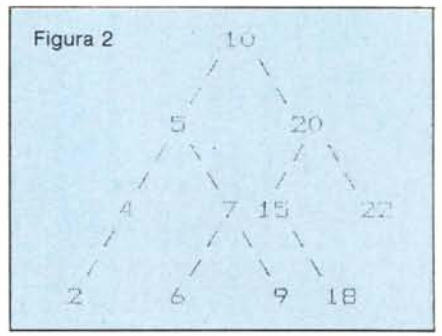


Gli algoritmi di inserzione e ricerca sono molto semplici: se il dato da introdurre (o trovare) è maggiore del dato con cui lo sto confrontando, continuo tra i figli di destra, se è minore tra i figli sinistri. Per esempio, se voglio introdurre «6» lo confronto col primo dato, «10».

«6» < «10», quindi vado a sinistra e trovo «5».

«6» > «5», quindi vado a destra e trovo «7».

«6» < «7». Il figlio Sn. di «7» non esiste, quindi la posizione è libera. L'albero adesso sarà:



```

1
1 'save "alber0"
60000 REM *** on... goto... ***
60010 HND=0
60050 ON HS GOSUB 60100,60200,60400,60600,60800,61000,61400,61600,61800,62000,61
200,62200
60070 HS=0:HSIC=0:HMD=0
60099 RETURN ***
60100 REM *** inizio ***
60120 DEFINT HD:IM HNR(2,15),H(3,15),H$(15)
60199 RETURN
60200 REM *** open ***
60230 HNR(1,HCI)=HLC:GOSUB 63000:SET # HCI,1:GOSUB 63800
60240 HNR(0,HCI)=H(0,HCI):HNR(2,HCI)=2
60399 RETURN
60400 REM *** inserimento ***
60410 GOSUB 60600
60420 IF HT THEN RETURN
60430 HUR=HUR+1:HNR(0,HCI)=HUR:HRD=HUR:IF HMD THEN HRD=HRD1:HMD=0
60440 IF HUR=2 THEN HPE=0:GOTO 60470
60450 IF HI THEN H(1,HCI)=HUR ELSE H(0,HCI)=HUR
60460 GOSUB 63700:PUT #HCI,HP
60470 H(0,HCI)=0:H(1,HCI)=0:H(2,HCI)=HPP:H(3,HCI)=HRD:H$(HCI)=HCH$
60480 GOSUB 63700:PUT #HCI,HUR
60490 IF HSIC THEN GOTO 60520
60500 GOSUB 63300
60520 HNR(2,HCI)=HUR
60599 RETURN
60600 REM *** ricerca ***
60603 IF LEN(HCH$)>HNR(1,HCI) THEN HCH$=LEFT$(HCH$,HNR(1,HCI))
60605 HPE=STRING$(HNR(1,HCI)-LEN(HCH$),32):HCH$=HCH$+HPE$
60610 HUR=HNR(0,HCI):HT=0
60620 IF HUR=1 THEN HPI=HUR:GOTO 60710
60640 HP=HP+1:SET #HCI,HP:GOSUB 63800:HCC$=H$(HCI)
60670 IF HCH$=HCC$ THEN HT=1:HRD=H(3,HCI):GOTO 60710
60680 IF HCH$>HCC$ THEN HI=1:HP=H(1,HCI) ELSE HI=0:HP=H(0,HCI)
60700 HPP=HP:IF HP1 THEN 60640
60710 RETURN
60799 RETURN
60800 REM *** cancella ***
60810 IF HT=0 THEN 60999
60815 HRD1=HRD
60820 IF HNR(0,HCI)=2 THEN 60910
60825 HPC$=H(0,HCI):HPCD$=H(1,HCI):HPPC=H(2,HCI):HPRC=H(3,HCI):HDC$=H$(HCI):HNDC=
HNR(2,HCI)
60830 IF HPDC=0 THEN 60925
60835 GET #HCI,HPCD:GOSUB 63800
60840 HPSFDC=H(0,HCI):HPPDFDC=H(1,HCI):HPPFDC=H(2,HCI):HPRFDC=H(3,HCI):HFDC$=H$(H
CI):HNDFDC=HPDC
60845 IF HPSFDC=0 THEN 60955
60850 HP=HNDFDC:HI=1
60855 GOSUB 63400
60860 HPS$=H(0,HCI):HPDS=H(1,HCI):HPRS=H(2,HCI):HPRS$=H$(HCI):HNDS$=
H(3,HCI):HNDS$=H(3,HCI):HNDS$=H$(HCI):HNDS$=
60865 IF HPDS=0 THEN 60980
60870 HI=0:HP=HP+1:GOSUB 63400
60875 HNDD=HP1
60880 H(1,HCI)=HPPDC:GOSUB 63700:PUT #HCI,HNDD
60885 GET #HCI,HPSD:GOSUB 63800:H(2,HCI)=HPPFDC:GOSUB 63700:PUT #HCI,HPDS
60890 GET #HCI,HPPS:GOSUB 63800:H(0,HCI)=0:GOSUB 63700:PUT #HCI,HPPS
60895 GET #HCI,HNDFDC:GOSUB 63800:H(2,HCI)=HNDD:GOSUB 63700:PUT #HCI,HNDFDC
60900 GET #HCI,HNDC:GOSUB 63800:H(1,HCI)=HPRS:H(3,HCI)=HPS:H$(HCI)=HS:GOSUB 63
700:PUT #HCI,HNDC
60905 HNDEL=HNDS:GOSUB 63900:GOTO 60995
60910 HNDEL=2:GOSUB 63900
60915 HNR(0,HCI)=1:GOSUB 63300
60920 HNR(2,HCI)=2:GOTO 60999
60925 IF HPS=0 THEN 60935
60930 GET #HCI,HPCD:GOSUB 63800:H(2,HCI)=HPPC:GOSUB 63700:PUT #HCI,HPSC
60935 GET #HCI,HPPC:GOSUB 63800
60940 IF HNDC=H(0,HCI) THEN H(0,HCI)=HPC ELSE H(1,HCI)=HPC
60945 HNDEL=HNDC:GOSUB 63900:PUT #HCI,HPPC
60950 HNDEL=HNDC:GOSUB 63900:HNR(2,HCI)=HPPC:GOTO 60999
60955 HPC$=HFDC$:HPRC=HPRFDC:HPDC=HPDFDC

```

```

3
61815 GOSUB 60200
61820 IF HOTT=0 THEN 61835
61825 OPEN "R",#HCFDS,HFDS$,HLFD:FIELD HCFDS,HLFD AS HF01$
61830 OPEN "R",#HCFDD,HFDD$,HLFD:FIELD HCFDD,HLFD AS HF02$
61835 HP2=1:WHILE HP2<HRA:HP2=HP2*2:WEND:HP2=HP2*2
61840 WHILE HP2<1
61845 HUR=HP2/2
61850 FOR HAY=HUR TO HRA STEP HP2
61855 GOSUB 62800
61860 GOSUB 60400
61865 IF HT THEN 61875
61870 IF HOTT THEN GOSUB 62900
61875 NEXT HX
61880 HP2=HP2/2
61885 WEND
61915 IF HOTT=0 THEN HNR(0,HCI)=HFIN
61916 GOSUB 61400
61917 CLOSE
61920 KILL HALF:KILL HFIS$
61925 IF HOTT THEN KILL HFDS$
61930 NAME HFIDE AS HFIDEF$
61935 IF HOTT THEN NAME HFDD$ AS HFDDF$
61999 RETURN
62000 REM *** inigen ***
62005 HNR(1,HCI)=HLC:CLOSE HCI:GOSUB 63000:HNR(0,HCI)=1:GOSUB 63300:CLOSE HCI
62199 RETURN
62200 REM *** stat ***
62220 HSF=0:HNUMC=0:HOTT=0:HRDE=0
62240 FOR HX=2 TO HNR(0,HCI)
62260 GET #HCI,HX:GOSUB 63800
62280 IF ASC(H$(HCI))=42 THEN HRDE=HRDE+1:GOTO 62310
62300 IF H(0,HCI)=0 XOR H(1,HCI)=0 THEN HSF=HSF+1
62310 NEXT HX
62320 HRD=HNR(0,HCI):HRU=HRD-HRDE:HOTT=100-(HSF*100/HRU):HDISK=(HNR(1,HCI)+2)
*HRU/1000:HDISK0=HDISK/1+(HRU*HLFD)/1000
62399 RETURN
62500 REM *** alfa ***
62510 HRA=1:HCI=HCIS:HFIS$=HFIS$
62520 GOSUB 60200
62530 OPEN "R",#HCA,HALF$,HLC+2:FIELD HCA,2 AS HRD2$,HLC AS HCHA$
62540 HI=0:GOSUB 61200
62560 GOSUB 62400:PUT #HCA,HRA
62570 GOSUB 61000
62580 IF HT THEN HRA=HRA+1:GOTO 62560
62590 CLOSE HCI
62599 RETURN
62600 REM *** putalfa ***
62610 LSET HRD2$=MKI$(H(3,HCI)):LSET HCHA$=H$(HCI)
62699 RETURN
62700 REM *** getalfa ***
62710 HRD1=CVI(HRD2$):HCH$=HCHA$
62799 RETURN
62800 REM *** transalfa ***
62810 GET #HCA,HXY:GOSUB 62700
62820 IF HOTT THEN HRD=0 ELSE HMD=1
62899 RETURN
62900 REM *** transdati ***
62910 GET #HCFDS,HRD1
62920 LSET HFDD$=HFDD1$
62930 PUT #HCFDD,HRD
62999 RETURN
63000 REM *** open1 ***
63010 OPEN "R",#HCI,HFIS$,HLC+8
63020 GOSUB 63200
63099 RETURN
63200 REM *** setind ***
63205 ON HCI GOTO 63210,63215,63220,63225,63230
63210 FIELD HCI,2 AS HFP51$,2 AS HFPD1$,2 AS HFP1$,2 AS HFR1$,HNR(1,HCI) AS HF
CI1:RETURN
63215 FIELD HCI,2 AS HFP52$,2 AS HFPD2$,2 AS HFP2$,2 AS HFR2$,HNR(1,HCI) AS HF
CI2:RETURN
63220 FIELD HCI,2 AS HFP53$,2 AS HFPD3$,2 AS HFP3$,2 AS HFR3$,HNR(1,HCI) AS HF
CI3:RETURN
63225 FIELD HCI,2 AS HFP54$,2 AS HFPD4$,2 AS HFP4$,2 AS HFR4$,HNR(1,HCI) AS HF
CI4:RETURN
63230 FIELD HCI,2 AS HFP55$,2 AS HFPD5$,2 AS HFP5$,2 AS HFR5$,HNR(1,HCI) AS HF

```

4

```

60960 IF HPDC=0 THEN 60970
60965 GET #HCI,HPDFDC:GOSUB 63800:H(2,HCI)=HPDFDC:GOSUB 63700:PUT #HCI,HPDFDC
60970 GET #HCI,HNDC:GOSUB 63800:H(1,HCI)=HNDC:H(3,HCI)=HFC:HF(HCI)=HDC#:GOSUB 6
3700:PUT #HCI,HNDC
60975 HNDEL=HNDFDC:GOSUB 63900:HNR(2,HCI)=HNDFDC:GOTO 60999
60980 GET #HCI,HPFS:GOSUB 63800:H(0,HCI)=0:GOSUB 63700:PUT #HCI,HPFS
60985 GET #HCI,HNDC:GOSUB 63800:H(3,HCI)=HNS:HF(HCI)=HNS#:GOSUB 63700:PUT #HCI,H
NDC
60990 HNDEL=HNMS:GOSUB 63900
60995 HNR(2,HCI)=HNDC
60999 RETURN
61000 REM *** Rc. succ. ***
61010 HT=1:HCR=HNR(2,HCI)
61020 GET #HCI,HCR:GOSUB 63800
61030 HIC=HI:GOSUB 63600
61040 IF HP THEN 61130
61050 HCC=HCR:HCR1=HCR
61060 HPAT=H(2,HCI)
61070 IF HPAT=0 THEN HCON=HCR1:GOTO 61150
61080 GET #HCI,HPAT:GOSUB 63800
61090 HIC=HI:GOSUB 63600
61100 IF HP=HCC THEN HCC=HPAT:GOTO 61060
61110 HCON=HPAT:GOTO 61150
61120 GOSUB 63400:HCON=HP1
61130 GOSUB 63500:HRO=H(3,HCI)
61199 RETURN
61200 REM *** basalt ***
61210 HP=2:HI=1-HI
61220 GOSUB 63400
61230 HRD=H(3,HCI)
61399 RETURN
61400 REM *** fine ***
61410 GOSUB 63300:CLOSE HCI
61599 RETURN
61600 REM *** modif. ***
61610 HMD=1
61620 HCH=HCHDF
61630 GOSUB 60800
61640 IF HT=0 THEN 61700
61650 HCH=HCHI#
61660 GOSUB 60400
61700 HMD=0
61799 RETURN
61800 REM *** ottim. ***
61802 HFIN=HNR(0,HCI)
61805 CLOSE
61810 GOSUB 62500
61812 HCI=HCID:HF1=HFID#:GOSUB 62000

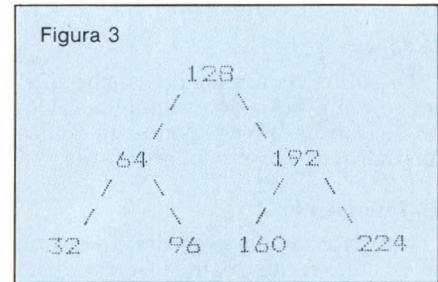
```

```

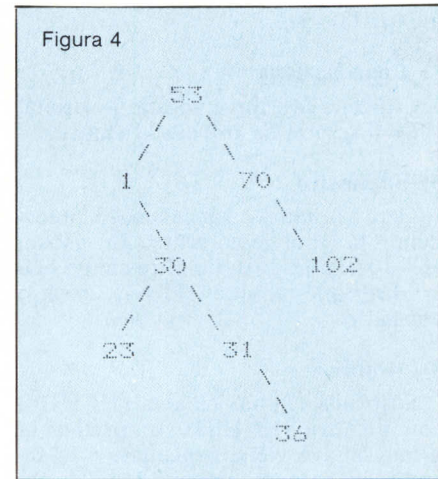
CH5#:RETURN
63299 RETURN
63300 REM *** snur ***
63320 H(0,HCI)=HNR(0,HCI)
63330 GOSUB 63700:PUT #HCI,1
63399 RETURN
63400 REM *** cbas ***
63410 WHILE HP<>0
63420 HP1=HP:GET #HCI,HP:GOSUB 63800
63430 HNR(2,HCI)=HP1
63440 HIC=1-HI:GOSUB 63600
63450 WEND
63499 RETURN
63500 REM *** contr ***
63510 IF HCON=HCR THEN HT=0:RETURN
63520 HCR=HCON:HNR(2,HCI)=HCR
63599 RETURN
63600 REM *** X ***
63610 IF HIC THEN HP=H(1,HCI) ELSE HP=H(0,HCI)
63699 RETURN
63700 REM *** putind ***
63705 ON HCI GOTO 63710,63715,63720,63725,63730
63710 LSET HFPS1#=#MKI#(H(0,1)):LSET HFDD1#=#MKI#(H(1,1)):LSET HFFP1#=#MKI#(H(2,1))
:LSET HFPR1#=#MKI#(H(3,1)):LSET HFCH1#=#H#(1):RETURN
63715 LSET HFPS2#=#MKI#(H(0,2)):LSET HFDD2#=#MKI#(H(1,2)):LSET HFFP2#=#MKI#(H(2,2))
:LSET HFPR2#=#MKI#(H(3,2)):LSET HFCH2#=#H#(2):RETURN
63720 LSET HFPS3#=#MKI#(H(0,3)):LSET HFDD3#=#MKI#(H(1,3)):LSET HFFP3#=#MKI#(H(2,3))
:LSET HFPR3#=#MKI#(H(3,3)):LSET HFCH3#=#H#(3):RETURN
63725 LSET HFPS4#=#MKI#(H(0,4)):LSET HFDD4#=#MKI#(H(1,4)):LSET HFFP4#=#MKI#(H(2,4))
:LSET HFPR4#=#MKI#(H(3,4)):LSET HFCH4#=#H#(4):RETURN
63730 LSET HFPS5#=#MKI#(H(0,5)):LSET HFDD5#=#MKI#(H(1,5)):LSET HFFP5#=#MKI#(H(2,5))
:LSET HFPR5#=#MKI#(H(3,5)):LSET HFCH5#=#H#(5):RETURN
63800 REM *** getind ***
63805 ON HCI GOTO 63810,63815,63820,63825,63830
63810 H(0,1)=CVI(HFPS1#):H(1,1)=CVI(HFDD1#):H(2,1)=CVI(HFFP1#):H(3,1)=CVI(HFPR1#)
):H#(1)=HFCH1#:RETURN
63815 H(0,2)=CVI(HFPS2#):H(1,2)=CVI(HFDD2#):H(2,2)=CVI(HFFP2#):H(3,2)=CVI(HFPR2#)
):H#(2)=HFCH2#:RETURN
63820 H(0,3)=CVI(HFPS3#):H(1,3)=CVI(HFDD3#):H(2,3)=CVI(HFFP3#):H(3,3)=CVI(HFPR3#)
):H#(3)=HFCH3#:RETURN
63825 H(0,4)=CVI(HFPS4#):H(1,4)=CVI(HFDD4#):H(2,4)=CVI(HFFP4#):H(3,4)=CVI(HFPR4#)
):H#(4)=HFCH4#:RETURN
63830 H(0,5)=CVI(HFPS5#):H(1,5)=CVI(HFDD5#):H(2,5)=CVI(HFFP5#):H(3,5)=CVI(HFPR5#)
):H#(5)=HFCH5#:RETURN
63900 REM *** pude1 ***
63910 H#(HCI)="*
63920 GOSUB 63700:PUT #HCI,HNDEL
63999 RETURN

```

Un albero è perfettamente bilanciato quando ogni nodo ha esattamente 2 figli. Es.:



Questo è un albero perfettamente bilanciato, mentre nella figura 4 abbiamo un albero estremamente sbilanciato:



Occorre notare che un albero può essere perfettamente bilanciato solo quando è costituito da  $(n+2) - 2$  elementi.

Se ha un numero diverso di elementi, fatalmente qualche nodo avrà, meno di due figli, pur essendo la distribuzione al massimo dell'efficienza.

### Uso di Alber0

Nel programma alber0 ogni elemento dell'albero contiene:

1. Puntatore al figlio Sn.
2. Puntatore al figlio Dx.
3. Puntatore al padre
4. Puntatore al record dati
5. La chiave.

L'uso di `albero0` avviene chiamando le diverse subroutine che lo costituiscono e passando loro i parametri opportuni.

### 1) Inizio

Si esegue quando si avvia il programma. Definisce tutte le variabili inizianti con «H» come intere e dimensiona le matrici usate dal programma.

### 2) Open

Di solito viene eseguito anche questa solo all'inizio del programma. Dopo le subroutine «Inigen» ed «Ottimizza» occorre però riaprire i file.

### 3) Inserimento

Inserisce la chiave voluta. In uscita, se la chiave che voglio inserire esiste già, porrà la variabile «HT»=1 (altrimenti HT=0).

### 4) Ricerca

Cerca la chiave voluta. In uscita, HT=0 se la chiave non esiste, altrimenti HT=1.

### 5) Cancellazione

Cancella la chiave voluta. In uscita, HT=0 se non ha trovato la chiave.

### 6) Successivo

Cerca la chiave successiva o precedente la chiave corrente. In uscita, HT=0 se non esistono altre chiavi nella direzione voluta. HT=1 in caso contrario.

### 8) Modifica

Sostituisce una chiave (HCHDS) con un'altra (HCHSS). In pratica la prima chiave viene cancellata e la seconda inserita, mantenendo uguale solo il puntatore al Record dati.

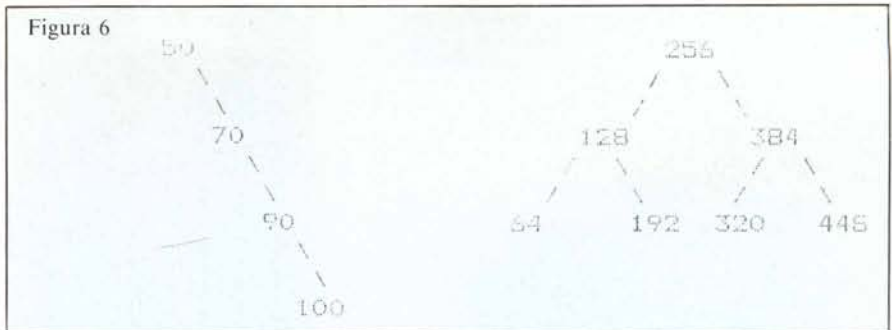
### 9) Ottimizzazione

Ottimizza il FI ed eventualmente anche il File Dati (FD).

Ciò avviene trasformando l'albero originario in un albero perfettamente bilanciato.

Se si ottimizza solo il FI il FD manterrà le stesse dimensioni (se esistevano degli spazi vuoti dovuti a cancellature restano), non solo, ma anche il volume del FI resta uguale. L'unica cosa che dovrebbe variare è la velocità di accesso.

È conveniente ottimizzare il FD quando ho eseguito molte cancellature ed ho problemi di spazio sul dischetto.



Per ottimizzare solo il FI devo fornire il nome del FI sorgente (FIS), quello del FI destinazione (FID), e quello del FI definitivo (FIDEF). Inoltre devo fornire il nome del File Alfabetico (FA).

Il FIS è il FI da ottimizzare. Il FID e il FA sono file temporanei che verranno cancellati alla fine della subroutine. Il FIDEF è il nome finale del FI, solitamente uguale a FIS.

Nel caso che voglia ottimizzare solo il FI bastano 2 canali, in quanto solo FIS e FA, e poi FA e FID sono aperti contemporaneamente.

Se invece voglio ottimizzare anche il FD devo fornire il nome del FD da ottimizzare (FDS), del FD destinazione (FDD) e il nome definitivo del FD (FDDEF).

Anche in questo caso il FDD è un file temporaneo.

Devo però poter aprire 4 file contemporaneamente. In un primo tempo ho aperti FIS e FA, poi FA, FID, FDS e FDD.

In qualsiasi caso posso aprire il FIS su qualsiasi canale, escluso il canale del FA.

La subroutine crea il FA leggendo in ordine alfabetico le chiavi del FIS nel FA. Quindi le chiavi del FA vengono inserite, in ordine opportuno, nel FID. Se il FD non viene ottimizzato il Puntatore al Record dati viene mantenuto, altrimenti viene ricalcolato.

### 10) Inigen

Inizializza e dimensiona i FI. Occorre farlo una volta sola per ogni FI.

### 11) Basalt

Cerca la chiave più alta o più bassa dell'albero.

### 12) Stat

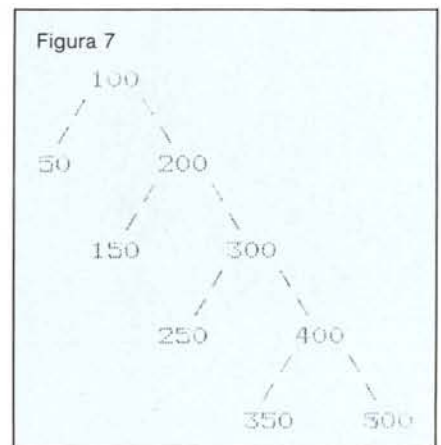
Dà una statistica dei FI.

Il valore «HOTT!» dà una misura percentuale dei nodi con un figlio solo. La percentuale di nodi con un solo figlio è legata al bilanciamento dell'albero.

Nella figura 6 l'albero a Sn. è assolutamente sbilanciato, e tutti i nodi

hanno un figlio solo. In un albero perfettamente bilanciato, tutti i nodi hanno due figli.

Attenzione, però: possono esistere alberi in cui tutti i nodi hanno due o nessun figlio e che sono sbilanciati:



In questo caso, nonostante l'albero sia molto sbilanciato, HOTT! sarà uguale al 100%.

### Avvertenze

`Albero0`, così com'è, può gestire fino a 5 FI contemporaneamente. Se si vogliono gestire altri FI (!), basta modificare le seguenti linee:

63205 ON HCI GOTO 63210, ..., 63230 aggiungendo un numero di linea (<63299) per ogni ulteriore file.

Es.: 63205 ON HCI GOTO 63210, ..., 63230, 63235, 63240 le linee aggiunte avranno la seguente struttura:

```

XXXXX FIELD HCI, 2 AS
HFSPSy$, 2 AS HFPDy$, 2 AS
HFPPy$, 2 AS HFPRy$, HNR
(1,HCI) AS HFCHy$ dove «y» è un
numero che contraddistingue le
variabili.

```

«y» deve essere uguale nella stessa linea e diverso da linea a linea.

Idem alla linea 63705:

```
63705 ON HCI GOTO ...
```

ed alla linea 63805.

Le linee aggiunte seguono il principio stabilito per la linea 63205.

MC

# I capolavori

# Star si adattano ad ogni cornice!



Solo con una stampante di elevata qualità si possono sfruttare ed esaltare tutti i vantaggi di un computer di elevata qualità. Star produce stampanti per computer che rispondono ad ogni tipo di esigenza. Ogni stampante Star è un capolavoro che associa robustezza a precisione, poichè Star è un maestro nella sua arte! Ogni capolavoro Star regge qualsiasi tipo di confronto in quanto a tecnologia, prestazione e prezzo. Belle o veloci, flessibili o specializzate, le stampanti Star si adattano ad ogni computer, poco importa come si chiami e dove si trovi! Dunque, non si affanni a cercare oltre, ma entri anche Lei nella galleria del nostro rivenditore di zona, dove troverà sicuramente la stampante che cerca, unitamente a tanti buoni consigli: siamo certi che anche Lei concluderà che, **con una Star si può andare molto lontano!**

**star**   
La tua stampante

 **DISTRIBUTORE PER L'ITALIA**  
**CITRON**<sub>spa</sub>  
Via Gallarate, 211 20151 Milano  
tel. 02/301.00.81 r.a. 301.00.91 r.a.

Per avere maggiori informazioni e l'indirizzo del rivenditore della Sua zona, ci invii il coupon allegato.

Ditta: \_\_\_\_\_ Via: \_\_\_\_\_ n° \_\_\_\_\_  
Nome: \_\_\_\_\_ Cap.: \_\_\_\_\_ Citta: \_\_\_\_\_  
Tel.: \_\_\_\_\_