

# PRAKTIKUM

R

O

G

R

A

# MERSKI

LJUBLJANA,

JUNIJ 1971

BATAGELJ VLADIMIR  
IV  
1970  
71  
tehnična matematika

NA NAPAKAH SE UČIMO !

naloge:

### N A J K R A J Š E   P O T I

ali:

kako sem zaradi

```
KAYEN(KOREN)=0
NAYEN(KOREN)=1
305 NSUMA=0
DO 306 J=1,NUMER
306 NSUMA=NSUMA+NAMEN(J)
IF(NSUMA) 312,312,307
307 DO 311 K=1,NUMER
IF(NAMEN(K)) 311,311,308
308 KEKEC=MOZEG(K+1)-MOZEG(K)
IF(KEKEC) 311,311,309
309 NAMEN(K)=0
KORAK=3*MOZEG(K)-4
DO 311 I=1,KEKEC
KORAK=KORAK+3
LUKEC=MREZA(KORAK)
KIPEC=MINKA(K)+MREZA(KORAK+1)
IF(KIPEC=MINKA(LUKEC)) 310,311,311
310 MINKA(LUKEC)=KIPEC
KAYEN(LUKEC)=K
NAMEN(LUKEC)=1
311 CONTINUE
GO TO 305
312 RETURN
```

izgubljal dneve in dneve ...

Pri predmetu "Programerski praktikum" sem si izbral in dobil v obdelavo naslednjo nalogu:

"/Minimalna pot/. Podano je cestno omrežje z zanimi dolžinami cest - spojnic med vozli. Za izbrane pare vozlov poišči najkrajšo pot skozi mrežo."

Spočetka sem mislil v tem uvodu razviti celo teorijo o ekstremnih poteh, vendar sem se kasneje premislil in odločil za naslednjih nekaj vrstic.

Označimo križišča z  $x_i$ , ceste z  $u_i$  in ustrezní množici z  $X$  in  $U$ . Vsaki cesti  $u_i$  pripisimo še neko nenegativno število  $d_i = d(u_i)$  - njeno dolžino. Zaporedju cest, ki prehajajo ena v drugo, recimo pot

$$s = (u_i)_{i \in \{1, n\}}$$

Če ima pot  $s$  začetek v križišču  $x_i$  in konec v križišču  $x_j$ , označimo to

$$s = s_{ij}$$

množico vseh takih poti pa s  $\mathcal{P}_{ij}$ . Poti, ki gre skozi vsako križišče največ enkrat, pravimo enostavna ali tudi aciklična pot. Množico vseh enostavnih poti iz  $x_i$  v  $x_j$  označimo  $\mathcal{P}^{e}_{ij}$ . Velja trditev: enostavna pot je sestavljena iz največ  $|X| - 1$  cest. Pojem dolžine lahko pospolimo na poti

$$d(s) = \sum_{i=1}^n d(u_i)$$

Sedaj pa že lahko definiramo pojem najkrajše poti. Bodи  $\mathcal{P}_{ij} \neq \emptyset$  potem je  $s_m$  najkrajša pot, če zadošča pogojem:

$$1. \quad s_m \in \mathcal{P}_{ij}$$

$$2. \quad d(s_m) = \min_{s \in \mathcal{P}_{ij}} \{d(s)\}$$

Ker je lahko množica  $\mathcal{P}_{ij}$  tudi neskončna bi se lahko zgodilo, da bi bil za določitev  $s_m$  potreben neskončen postopek. Da temu ni tako nam pove naslednja trditev:

Tl.

$$S_m \in \mathcal{P}_{ij}^{(k)}$$

Množica  $\mathcal{P}_{ij}^{(k)}$  pa je za omrežje s končnim številom cest in križišč tudi sama končna.

Označimo s  $\mathcal{P}_{ij}^{(k)}$  množico vseh poti iz  $x_i$  v  $x_j$ , ki so sestavljene iz kvečjemu  $k$  cest. Velja trditev:

$$T2. \min_{S \in \mathcal{P}_{ij}^{(k)}} \{d(S)\} = \min_{S \in \mathcal{P}_{ij}^{(k)}} \{d(S)\} \Rightarrow S_m \in \mathcal{P}_{ij}^{(k)}$$

Naslednja trditev je precej trivialna, vendar zato nič manj pomembna:

T3. Poljubna podpot najkrajše poti je najkrajša med potmi z istimi krajišči.

Toliko splošne filozofije o najkrajših poteh. Sledi kratek opis programa.

#### Vhodni podatki

ime	tip	vrsta	komentar
MAJOR	IN	VA	število cest
NUMER	IN	VA	število križišč
NICLA	IN	VA	nevtral za operacijo min - teoretično v praksi neko zadosti veliko število
KOPER	IN	VA	število začetnih križišč
IZOLA	IN	VA	število končnih križišč
MREZA	IN	AR	vektor sestavljen iz trojic MREZA( $3*k+1$ ) - začetek $k$ -te ceste MREZA( $3*k+2$ ) - konec $k$ -te ceste MREZA( $3*k+3$ ) - dolžina $k$ -te ceste
KRIZ1	IN	AR	začetna križišča
KRIZ2	IN	AR	končna križišča

Opomba: v našem programu veljajo omejitve  $MAJOR \leq 1200$ ,  $NUMER \leq 300$  in križišča označujemo z naravnimi števili od 1 do 300.

Iškanje najkrajše poti je razdeljeno na tri podprograme. Oglejmo si kaj naredijo.

REDAR - uredi trojke v vektorju MA po velikosti prvih komponent. Poleg tega nam MIND pove razporejenost podatkov o cestah z istim začetkom. Tako podatke o cestah z začetkom I vsebujejo MIND(I) - ta do MIND(I+1) - 1 - ta trojka.

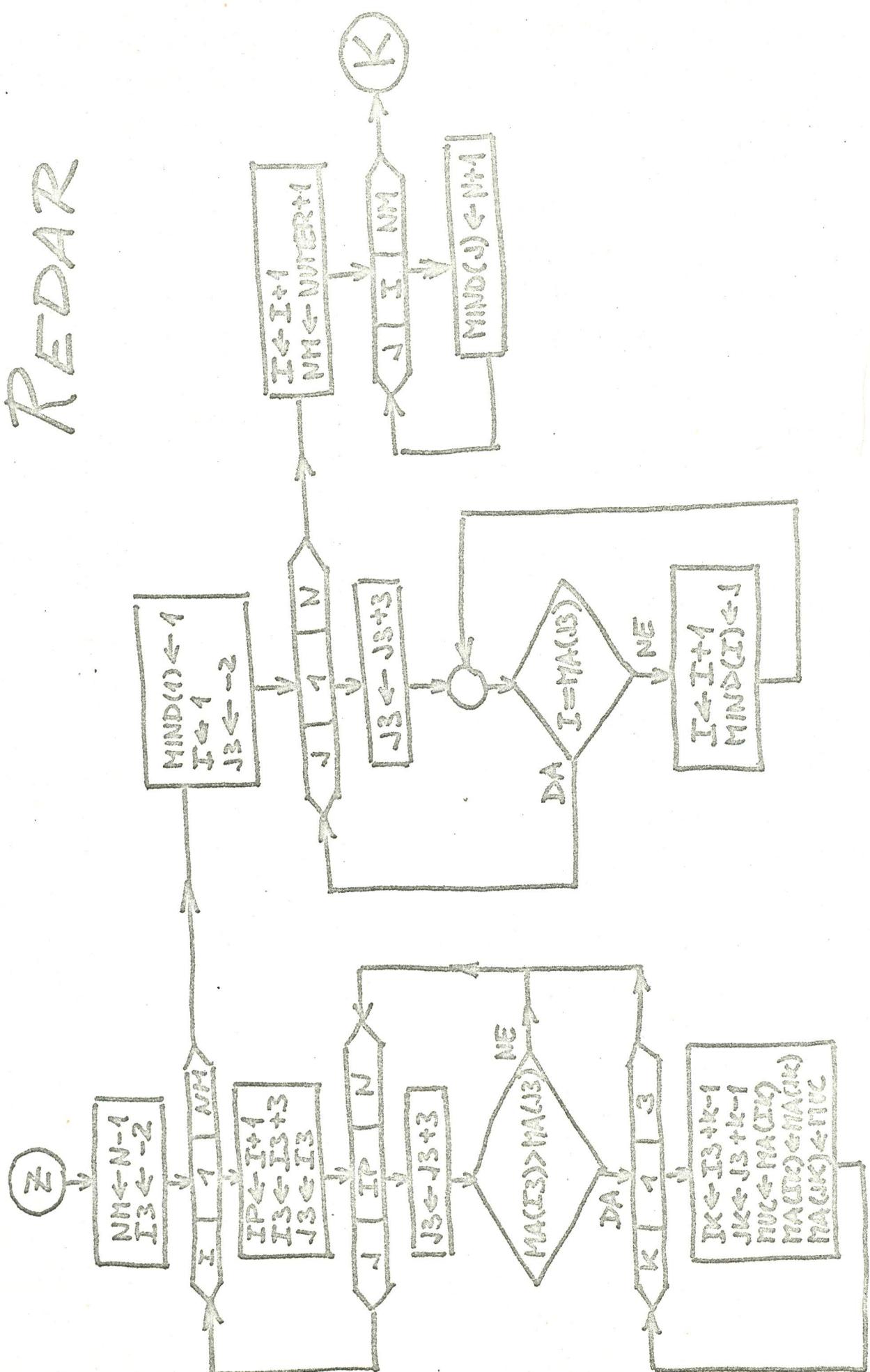
ROMAR - predpostavlja, da je vektor MREZA urejen po velikosti prvih komponent trojic podatkov in, da je znan vektor razporejenosti MOZEG . To dosežemo tako, da podatke čitamo že urejene in čitamo kot podatek tudi vektor MOZEG , ali pa da nad vektorjem MREZA in MOZEG uporabimo podprogram REDAR. V prvem primeru je seveda REDAR odveč. ROMAR nam za dano začetno križišče KOREN določi vektorja MINKA in KAMEN . MINKA( $I$ ) je enaka dolžini najkrajše poti KOREN -  $I$  , če je enaka NICLA pot ne obstaja. KAMEN( $I$ ) =  $K$  - pri zadnji minimizaciji poti KOREN -  $I$  smo prišli v  $I$  iz  $K$  /uporaba T3./. Če je KAMEN( $I$ ) = 0 , pot ne obstaja. Zaradi T2. se proces ustavi, ko je NSUMA = 0 in je dobijena pot res najkrajša.

CESTA - iz vektorjev KAMEN in MINKA določi z uporabo T3.  
najkrajšo pot KOREN - LIST in njeno dolžino.

Na naslednjih straneh so nанизani diagrami poteka vseh treh podprogramov in njih FORTRAN-ska realizacija. Sledi glavni program in stran rezultatov pri podatkih

kjer si v prvi vrstici sledi količine MAJOR,NUMER,NICLA,KOPER in IZOLA.Naslednje štiri vrstice zavzema že urejena MREZA ,v zadnji pa je izpisani MOZEG .Za stranjo rezultatov sem dodal še ALGOL-sko izvedbo programa in stran rezultatov za neko drugo cestno omrežje.

## REDA R



PAGE 3 E151 10 UREJANJE PODATKOV

```

SUBROUTINE REDAR(MA,MIND,NUMER,N)
DIMENSION MA(3600),MIND(301)
NM=N-1
I3=-2
DO 663 I=1,NM
IP=I+1
I3=I3+3
J3=I3
DO 663 J=IP,N
J3=J3+3
IF (MA(I3)=MA(J3)) 663,663,661
661 DO 662 K=1,3
IK=I3+K-1
JK=J3+K-1
YUC=MA(IK)
MA(IK)=MA(JK)
662 MA(JK)=YUC
663 CONTINUE
MIND(1)=1
I=1
J3=-2
DO 666 J=1,N
J3=J3+3
664 IF (I=MA(J3)) 665,666,665
665 I=I+1
MIND(I)=J
GO TO 664
666 CONTINUE
I=I+1
NM=NUMER+1
DO 667 J=I,NM
667 MIND(J)=N+1
RETURN
END

```

## FEATURES SUPPORTED

ONE WORD INTEGERS

## CORE REQUIREMENTS FOR REDAR

COMMON 0 VARIABLES 12 PROGRAM 260

END OF COMPIRATION

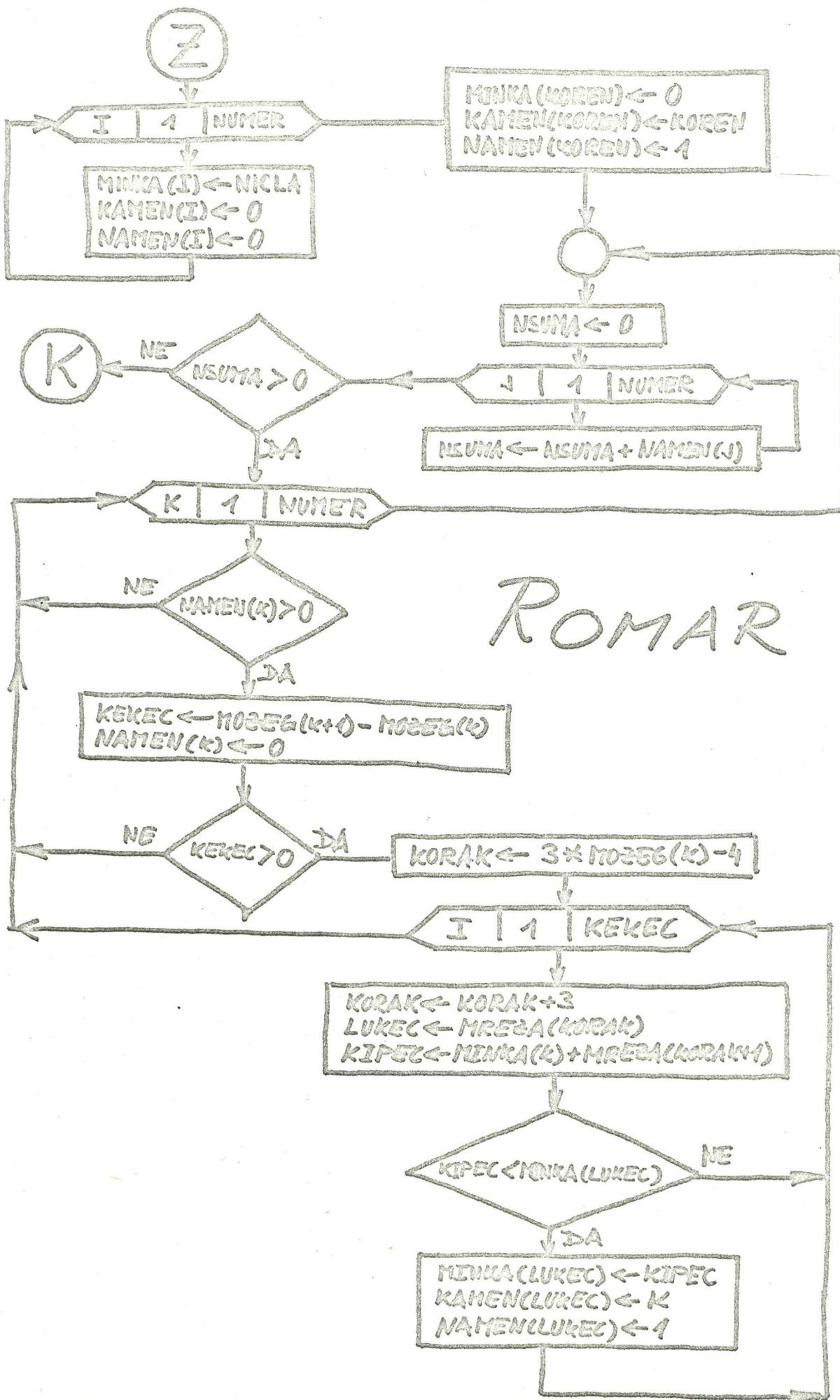
// DUP

```

*STORE WS UA REDAR
CART ID 0001 DB ADDR 2784 OB CNT 0010

```

// EJECT



PAGE 5 E151 10 MAKRAJE POTI IZ KRIZISCA KOREN

SUBROUTINE ROMAR(MREZA,MICLA,KOREN,VOZEC,IMKA,KAYEN)

DIMENSION MREZA(3000),MOZEG(301),MINKA(300),KAMEN(300),NAMEV(300)

NPITE(3,303) KOREN

303 FORMAT(1H //,20X,MAKRAJE POTI IZ KRIZISCA!,14)

DO 304 I=1,NUMER

MINKA(I)=MICLA

KAYEN(I)=0

304 NAMEV(I)=0

MINKA(KOREN)=0

KAYEN(KOREN)=KOREN

NAMEV(KOREN)=1

305 NSUMA=0

DO 306 J=1,NUMER

306 NSUMA=NSUMA+MAKEU(J)

IF(NSUMA).LT.307,307,300

307 RETURN

308 DO 313 K=1,NUMER

IF(NAMEC(K)).LT.313,313,310

309 KEKEC=MOZEG(K+1)-MOZEG(K)

NAMEV(K)=0

IF(KEKEC).LT.313,313,310

310 KORAK=3\*MOZEG(K)+4

DO 312 I=1,KEKEC

KORAK=KORAK+3

LUKEC=MREZA(KORAK)

KIPEC=MINKA(K)+MREZA(KORAK+1)

IF(KIPEC-MINKA(LUKEC)).LT.311,312,312

311 MINKA(LUKEC)=KIPEC

KAMEN(LUKEC)=K

NAMEV(LUKEC)=1

312 CONTINUE

313 CONTINUE

GO TO 305

END

## FEATURES SUPPORTED

ONE WORD INTEGERS

## CORE REQUIREMENTS FOR POLAR

COMMON 0 VARIABLES 312 PROGRAM 298

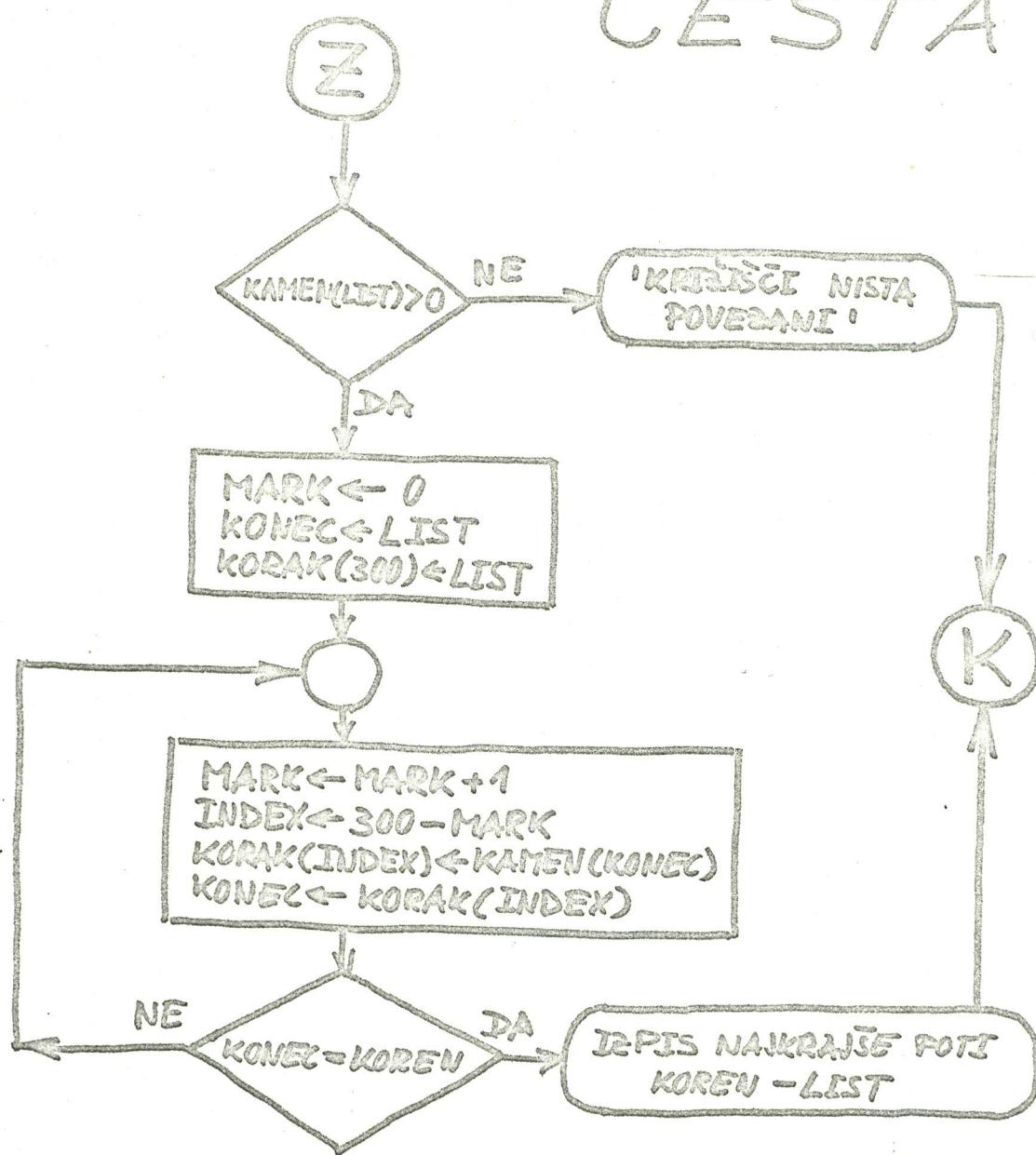
END OF COMPIRATION

// DUP

\*STORE WS UA ROMAR  
CART ID 0001 DB ADDR 2799 DB CNT 0013

// EJECT

# CESTA



PAGE 7 E151 : 10 MAJKRAUSA POT KOREN = LIST

SUPPORTING CESTA(KOREN,LIST,KAMEN,MINKA)  
DIMENSION KAMEN(300),MINKA(300),KORAK(300)  
IF(KAMEN(LIST)) 996,996,291  
291 MARK=0  
KONEC=LIST  
KORAK(300)=LIST  
992 MARK=MARK+1  
INDEX=300-MARK  
KORAK(INDEX)=KAMEN(KONEC)  
KONEC=KORAK(INDEX)  
IF(KONEC-KOREN) 992,993,992  
993 WRITE(3,994) KOREN,LIST,MINKA(LIST)  
994 FORMAT(1H //,10X,'MAJKRAUSA POT IZ KRIZISCA',I4,' V KRIZISCE',I4,  
\*' IMA DOLZINO',I5//,10X,'IN GRE SKOZI NASLEDNJA KRIZISCA',//)  
WRITE(3,995) (KORAK(I),I=INDEX,300)  
995 FORMAT(/,10X,15(I3,,'))  
RETURN  
996 WRITE(3,997) LIST,KOREN  
997 FORMAT(1H //,10X,'KRIZISCE',I4,' NI DOSEGLJIVO IZ KRIZISCA',I4)  
RETURN  
END

FEATURES SUPPORTED

ONE WORD INTEGERS

CORE REQUIREMENTS FOR CESTA

COMMON 0 VARIABLES 306 PROGRAM 228

END OF COMPIILATION

// DUP

\*STORE WS UA CESTA  
CART ID 0001 DB ADDR 27AB DB CNT 000F

// EJECT

PAGE 1 E151 10

// JOP T

E151 10 DATACELL

LOC DRIVE	CART SPEC	CART AVAIL	FPUY DRIVE
0000	0001	0001	0000
		0002	0001

V2 M07 ACTUAL 16K COMPACT 16K

// XEQ ZX

// EJECT

// FOR

\*\* UREJANJE PODATKOV  
\* NAME REDAR  
\* ONE WORD INTEGERS  
\*LIST SOURCE PROGRAM

{ REDAR }

// FOR

\*\* NAJKRAJSE POTI IZ KRIZISCA KOTEN  
\* NAME ROMAR  
\* ONE WORD INTEGERS  
\* LIST SOURCE PROGRAM

{ ROMAR }

// FOR

\*\* NAJKRAJSA POT KOREN - LIST  
\* NAME CESTA  
\* ONE WORD INTEGERS  
\*LIST SOURCE PROGRAM

{ CESTA }

PAGE 8 E151 10

```
// FOR
*ONE WORD INTEGERS
*ICCS(1132 PRINTER,CARD)
* LIST SOURCE PROGRAM
      DIMENSION MREZA(3600),MOZEG(301),MINKA(300),KAMEN(300),KRIZI(300),
      *KRIZZ(300)
      READ(2,881) MAJOR,NIJER,NICLA,KOPER,IZOLA
881 FORMAT(5I5)
      MTRI=3*MAJOR
      READ(2,882) (MREZA(I),I=1,MTRI)
882 FORMAT(7(13,13,14))
      READ(2,883) (KRIZI(I),I=1,KOPER)
      READ(2,883) (KRIZZ(I),I=1,IZOLA)
883 FORMAT(24I3)
      WRITE(3,884)
884 FORMAT(1H1,20X,'NAJKAJSE POTI MED IZBRANIMI KRIZISCI',//)
      CALL REDAR(MREZA,MOZEG,NUMBER,MAJOR)
      DO 885 I=1,KOPER
      KOREN=KRIZI(I)
      CALL ROMARI(MREZA,NUMBER,NICLA,KOREN,MOZEG,MINKA,KAMEN)
      DO 885 J=1,IZOLA
      LIST=KRIZZ(J)
885 CALL CESTA(KOREN,LIST,KAMEN,MINKA)
      CALL EXIT
      END
```

FEATURES SUPPORTED

ONE WORD INTEGERS  
ICCS.

CORE REQUIREMENTS FOR

COMMON 0 VARIABLES 5112 PROGRAM 220

END OF COMPILATION

// XEO

NAJKRAJSE POTI MED IZBRANIMI KRIZISCI

NAJKRAJSE POTI IZ KRIZISCA 8.

NAJKRAJSA POT IZ KRIZISCA 8 V KRIZISCE 7IMA DOLZINO 21  
IN GRE SKOZI NASLEDNJA KRIZISCA

8, 12, 9, 5, 6, 7,

NAJKRAJSA POT IZ KRIZISCA 8 V KRIZISCE 8IMA DOLZINO 0  
IN GRE SKOZI NASLEDNJA KRIZISCA

8, 9,

NAJKRAJSA POT IZ KRIZISCA 8 V KRIZISCE 12IMA DOLZINO 5  
IN GRE SKOZI NASLEDNJA KRIZISCA

8, 12,

NAJKRAJSA POT IZ KRIZISCA 8 V KRIZISCE 14IMA DOLZINO 44  
IN GRE SKOZI NASLEDNJA KRIZISCA

8, 12, 9, 5, 6, 1, 4, 10, 11, 13, 14,

NAJKRAJSA POT IZ KRIZISCA 8 V KRIZISCE 11IMA DOLZINO 39  
IN GRE SKOZI NASLEDNJA KRIZISCA

8, 12, 9, 5, 6, 1, 4, 10, 11,

NAJKRAJSE POTI IZ KRIZISCA 1

NAJKRAJSA POT IZ KRIZISCA 1 V KRIZISCE 7IMA DOLZINO 13  
IN GRE SKOZI NASLEDNJA KRIZISCA

1, 2, 3, 6, 7,

NAJKRAJSA POT IZ KRIZISCA 1 V KRIZISCE 8IMA DOLZINO 18  
IN GRE SKOZI NASLEDNJA KRIZISCA

1, 4, 10, 11, 13, 14, 8,

NAJKRAJSA POT IZ KRIZISCA 1 V KRIZISCE 12IMA DOLZINO 23  
IN GRE SKOZI NASLEDNJA KRIZISCA

1, 4, 10, 11, 13, 14, 9, 12,

PAGE 1 E151 10

// JOB T

E151 10 BATAGELJ

LOG DRIVE	CART SPEC	CART AVAIL	PHY DRIVE
0000	0001	0001	0000
		0002	0001

V2 M07 ACTUAL 16K CONFIG 16K

// XEQ ZX

// XEQ ALGOL

\* LIST

\* NAME NAJKRAJSA POT MED KRIZISCHEMA

\* PRINT SYMBOL TABLE


  
 ALGOLSKI PROGRAM

## IDENTIFIER LIST PRINTOUT

DELTA	IDENTIFIER	FPNR	MODE	TYPE	KIND	ADDR	ABL	MBL	DIM	ASSI	AL
42	* BB						0	0	0	139	
43	RED		NL	IN	VA	02F0	0	0	0	0	0
44	NUMER		NL	IN	VA	02F1	0	0	0	0	0
45	ROMAR		NL		PR	0363	0	1	7	0	0
46	* PB						0	0	0	75	
47	MREZA	0	FL	IN	AR		0	0	1	0	0
48	NUMER		NL	IN	VA	02F2	0	0	0	0	0
49	NICLA		NL	IN	VA	02F5	0	0	0	0	0
50	KOREN		NL	IN	VA	02F8	0	0	0	0	0
51	MOZEG	4	FL	IN	AR		0	0	1	0	0
52	MINKA	5	FL	IN	AR		0	0	1	0	0
53	VAMEN										

NO ERRORS DETECTED IN THIS COMPILATION

CORE REQUIREMENTS

I/O-BUFFERS 0221

CONSTANTS 0011 VARIABLES 0107

PROGRAM 157

END OF COMPILATION

// XEQ

## NAJKRAJSA POT MED KRIZISCEMA

LINE ALGOL SOURCE PROGRAM

1130 ALGOL V2 NOI

```

0001 'BEGIN'
0002 'INTEGER' RED,NUMER,,,
0003 'PROCEDURE' ROMAR(MREZA,NUMER,NICLA,KOREN,MOZEG,MINKA,KAMEN),,
0004   'VALUE' NICLA,NUMER,KOREN,, 'INTEGER' NICLA,NUMER,KOREN,,,
0005   'INTEGER' 'ARRAY' MREZA,MOZEG,MINKA,KAMEN,,,
0006 'BEGIN'
0007   'INTEGER' NSUMA,KORAK,MIHEC,JAKEC,I,J,K,,,
0008     'INTEGER' 'ARRAY' NAMEN(/1..NUMER/),,
0009     'FOR' I.=1 'STEP' 1 'UNTIL' NUMER 'DO'
0010     'BEGIN'
0011       KAMEN(/I/).=NAMEN(/I/).=0,,MINKA(/I/).=NICLA
0012     'END',,
0013     MINKA(/KOREN/).=0,,KAMEN(/KOREN/).=KOREN,,NAMEN(/KOREN/).=1,,,
0014 SKAVT,,,
0015   NSUMA.=0,,,
0016   'FOR' J.=1 'STEP' 1 'UNTIL' NUMER 'DO'
0017     NSUMA.=NSUMA+NAMEN(/J/),,
0018     'IF' NSUMA 'GREATER' 0 'THEN'
0019     'BEGIN'
0020       'FOR' K.=1 'STEP' 1 'UNTIL' NUMER 'DO'
0021       'BEGIN'
0022         'IF' NAMEN(/K/) 'GREATER' 0 'THEN'
0023         'BEGIN'
0024           'IF' MOZEG(/K+1/) 'GREATER' MOZEG(/K/) 'THEN'
0025           'BEGIN'
0026             NAMEN(/K/).=0,,KORAK.=3*MOZEG(/K/)-4,,,
0027             'FOR' I.=MOZEG(/K/) 'STEP' 1 'UNTIL' MOZEG(/K+1/)-1 'DO'
0028             'BEGIN'
0029               KORAK.=KORAK+3,,MIHEC.=MREZA(/KORAK/),,
0030               JAKEC.=MINKA(/K/)+MREZA(/KORAK+1/),,
0031               'IF' JAKEC 'LESS' MINKA(/MIHEC/) 'THEN'
0032               'BEGIN'
0033                 MINKA(/MIHEC/).=JAKEC.,,
0034                 KAMEN(/MIHEC/).=K,,NAMEN(/MIHEC/).=1
0035               'END'
0036             'END'
0037           'END'
0038           'ELSE'
0039             NAMEN(/K/).=0
0040           'END'
0041         'END',,
0042         'GO TO' SKAVT
0043       'END'
0044     'END',,
0045   'PROCEDURE' REDAR(MREZA,MOZEG,RED,NUMER),,
0046   'VALUE' RED,NUMER,, 'INTEGER' RÉD,NUMER,,,
0047   'INTEGER' 'ARRAY' MREZA,MOZEG,,,
0048   'BEGIN'
0049     'INTEGER' KORAK,SEDLO,PED,FLIP,FLOP,IZMIK,I,J,K,,,
0050     KORAK.=-2,,,
0051     'FOR' I.=1 'STEP' 1 'UNTIL' RED-1 'DO'
0052     'BEGIN'
0053       SEDLO.=I+1,,KORAK.=KORAK+3,,PED.=KORAK,,,
0054       'FOR' J.=SEDLO 'STEP' 1 'UNTIL' RED 'DO'
0055       'BEGIN'
0056         PED.=PED+3,,,
0057         'IF' MREZA(/KORAK/) 'GREATER' MREZA(/PED/) 'THEN'
0058         'FOR' K.=1 'STEP' 1 'UNTIL' 3 'DO'
0059         'BEGIN'
0060           FLIP.=KORAK+K-1,,FLOP.=PED+K-1,,IZMIK.=MREZA(/FLIP/),,

```

MREZA(/FLIP/).=MREZA(/FLOP/), MREZA(/FLOP/).=IZMIK

0061

```

0062      'END'
0063      'END'
0064      'END'.,
0065      MOZEG(/1/).=1., I.=1., PED.=-2.,,
0066      'FOR' J.=1 'STEP' 1 'UNTIL' RED 'DO'
0067      'BEGIN'
0068      PED.=PED+3.,,
0069 CAJZLC..,
0070      'IF' I 'NOT EQUAL' MREZA(/PED/) 'THEN'
0071      'BEGIN'
0072          I.=I+1., MOZEG(/I/).=J., 'GO TO' CAJZLC
0073      'END'.,
0074      'END'.,
0075      'FOR' J.=I+1 'STEP' 1 'UNTIL' NUMER+1 'DO'
0076      MOZEG(/J/).=RED+1.,,
0077      'END'.,
0078 'PROCEDURE' CESTA(KOREN,LIST,NUMER,KAMEN,MINKA),,
0079     'VALUE' KOREN,LIST,NUMER.,
0080     'INTEGER' KOREN,LIST,NUMER.,
0081     'INTEGER' !ARRAY! KAMEN,MINKA.,
0082     'BEGIN'
0083         'INTEGER' MARK,KONEC,INDEX.,
0084         'INTEGER' !ARRAY! KORAK(/1..NUMER/),,
0085         'IF' KAMEN(/LIST/) 'GREATER' 0 'THEN'
0086         'BEGIN'
0087             MARK.=1.,KONEC.=LIST.,,
0088 CIKEL..,
0089             KORAK(/MARK/).=KONEC.,MARK.=MARK+1.,KONEC.=KAMEN(/KONEC/),,
0090             'IF' KONEC 'NOT EQUAL' KAMEN(/KONEC/) 'THEN' 'GO TO' CIKEL.,
0091             KORAK(/MARK/).=KONEC.,,
0092             SYSACT(1,6,80),,SYSACT(1,14,3),,SYSACT(1,2,5),,
0093             OUTSTRING(1,'('NAJKRAJSA POT IZ KRIZISCA')'),,
0094             OUTINTEGER(1,KOREN),,OUTSTRING(1,'('V KRIZISCE')'),,
0095             OUTINTEGER(1,LIST),,OUTSTRING(1,'('IMA DOLZINO')'),,
0096             OUTINTEGER(1,MINKA(/LIST/)),,SYSACT(1,14,1),,SYSACT(1,2,5),,
0097             OUTSTRING(1,'('IN GRE SKOZI NASLEDNJA KRIZISCA')'),,
0098             SYSACT(1,14,2),,
0099             'FOR' INDEX.=MARK 'STEP' -1 'UNTIL' 1 'DO'
0100             OUTINTEGER(1,KORAK(/INDEX/)),,
0101             'END'
0102             'ELSE'
0103             'BEGIN'
0104                 SYSACT(1,14,3),,SYSACT(1,2,5),,OUTSTRING(1,'('KRIZISCE')'),,
0105                 OUTINTEGER(1,LIST),,
0106                 OUTSTRING(1,'('NI DOSEGLJIVO IZ KRIZISCA')'),,
0107                 OUTINTEGER(1,KOREN),,
0108             'END'
0109             'END',,
0110             ININTEGER(0,RED),,ININTEGER(0,NUMER),,
0111             'BEGIN'
0112                 'INTEGER' NICLA,KOREN,LIST,MEJA,INDEX.,
0113                 'INTEGER' !ARRAY! MREZA(/1..3*RED/),MOZEG(/1..NUMER+1/),
0114                 MINKA(/1..NUMER/),KAMEN(/1..NUMER/),,
0115                 ININTEGER(0,NICLA),,ININTEGER(0,MEJA),,
0116                 SYSACT(0,10,1),,INARRAY(0,MREZA),,
0117                 REDAR(MREZA,MOZEG,RED,NUMER),,SYSACT(1,15,1),,
0118                 'FOR' INDEX.=1 'STEP' 1 'UNTIL' MEJA 'DO'
0119                 'BEGIN'
0120                     ININTEGER(0,KOREN),,ININTEGER(0,LIST),,
0121                     ROMAR(MREZA,NUMER,NICLA,KOREN,MOZEG,MINKA,KAMEN),,
0122                     CESTA(KOREN,LIST,NUMER,KAMEN,MINKA),,

```

NAJKRAJSA POT MED KRIZISCIMA

LINE ALGOL SOURCE PROGRAM

1130 ALGOL V2 M01

0123 'END'  
0124 'END'  
0125 'END'

## REZULTATI

NAJKRAJSA POT IZ KRIZISCA 13 V KRIZISCE 6 IMA DOLZINO 122  
IN GRE SKOZI NASLEDNJA KRIZISCA

13 8 9 3 19 18 14 7 4 6

KRIZISCE 7 NI DOSEGLOIVO IZ KRIZISCA 24

NAJKRAJSA POT IZ KRIZISCA 10 V KRIZISCE 12 IMA DOLZINO 91  
IN GRE SKOZI NASLEDNJA KRIZISCA

10 7 2 1 12

NAJKRAJSA POT IZ KRIZISCA 16 V KRIZISCE 11 IMA DOLZINO 145  
IN GRE SKOZI NASLEDNJA KRIZISCA

16 17 5 15 14 7 4 6 11

NAJKRAJSA POT IZ KRIZISCA 11 V KRIZISCE 5 IMA DOLZINO 106  
IN GRE SKOZI NASLEDNJA KRIZISCA

11 6 4 7 14 15 5

77 \*

Za konec pa bi nakazal še drugo rešitev problema najkrajših poti. Nanjo me je spomnil Professor Branestawm ali po naše Profesor Modrinjak; razberemo jo iz naslednjega razgovora Profesorja Modrinjaka z njegovim prijateljem Colonelom:

"Listen", went on the Professor, "and I will explain."

The Colonel sat down on the garden roller and started listening.

"If you travel by coach from this town to the next it takes two hours," said the Professor. "But if you go twice as fast it takes only one hour."

"Of course," said the Colonel.

"And if you go twice as fast as that it takes only half an hour."

"Quite," said the Colonel.

"And if you go fast enough it takes no time at all, so that you get there the moment you start. Very well" - ...

Kakor vidimo iz gornjega, lahko potem takem problem rešimo tudi tako, da zgradimo vozilo, ki bo šlo zadosti hitro...

P.S. Še sreča, da ne verjamem Zenonu Aleatskemu.