

パズル「カツクロ」の難易度について

SATO, Kingo / 佐藤, 金吾

(出版者 / Publisher)

法政大学多摩研究報告編集委員会

(雑誌名 / Journal or Publication Title)

Hosei University Tama bulletin / 法政大学多摩研究報告

(巻 / Volume)

19

(開始ページ / Start Page)

39

(終了ページ / End Page)

68

(発行年 / Year)

2004-03-30

(URL)

<https://doi.org/10.15002/00003063>

パズル「カツクロ」の難易度について

佐藤金吾

On characteristic of difficult degree to solve Katukuro-puzzles

Kingo SATO

1. はじめに

数理パズルで用いられる論理力の指標化を目指し、すでにいくつかのパズルにつき解法の難易度について考察してきたが、この小論では「カツクロ」の難易度を扱う。

カツクロとは、右図のようなマス目状の四角形内に、表示されている数字を手がかりに、つぎの3つのルールに従って数字を入れるパズルである。

1) 盤面に出ている数は、その右 (or 下) の区切られた一列 (以下、区切り列と呼ぶ) に入る数字の合計を示す。

2) 使う数字は1から9まで。

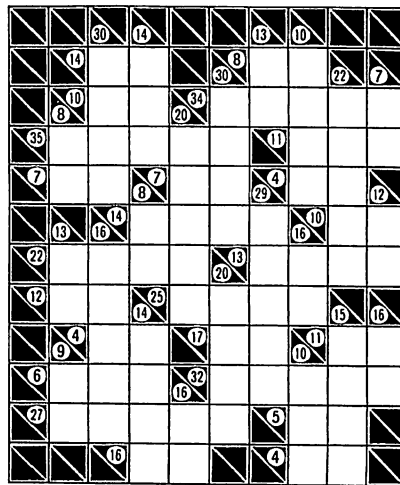
3) 区切り列に同じ数字は入らない。

なお、マスに入る数字を確定数と呼ぶ。

さて、以下の議論で必要なことを2つあげる。

① 実際の問題についての解の結果が必要なもので、問題例として3冊の問題集 [2] を用いた。

② 本文中のプログラムには、言語として「構造化 BASIC98」を用いた。



2. カツクロの解き方

2.1 基本的な解法

カツクロを解く一番の手がかりは、その合計数を一意に決める「数字の組」(この組以外にな

い) である。

例えば、区切り列の長さが3でその合計が23なら、6,8,9がこの3つのマスに入る一意に決まる‘数字の組’である。

(定義 2.1) 与えられた合計数から一意に決まるこの‘数字の組’を一意数群と呼ぶ。また、一意数群を持つ区切り列の中の各マスを一群マスと呼ぶ。

(定義 2.2) ある空きマスにおいて、①その中に入り得る数字の集まりで、確定数を含むものを可能数リスト、②まわりの状況等の条件からそのマスに入れない数字を除外数、と呼ぶ。なお、一群マスではその一意数群が可能数リストとなる。

(注意 1) どのマスも可能数リストを持つ。例えば、 $\{1, 2, \dots, 9\}$ を取ればよいが、要素が少ないものほどよい。

1) この「一意数群」だけを使う解法の手順を示す。

【スタート】与えられた合計数のすべてについて、一意数群を求める処理を行う。

(人がやる時には、問題集に付随している一意数群の表を参照する)

【手順 1】(数字の確定 1)

交差するタテとヨコの区切り列が共に一意数群を持つマスにおいて、両方の一意数群に含まれる数字でダブルものが一つなら、ダブル数字はその確定数となる。

【手順 2】(数字の確定 2)

ある一群マスにおいて、

- ① その一意数群から除外数を取り去った結果として一つの数字が残ったら、残った数字はその確定数となる。
- ② その一意数群の中のある数字が、区切り列の他のどのマスでも除外数となるなら、その確定数となる。

【手順 3】(区切り列の更新とそれに伴う処理)

あるマスで数字が確定したら、そのマスを区切り列から取り除く。すなわち、タテ、ヨコとも残った空きマス列を新しい区切り列とし、合計もそれに合わせる。また、元が一意数群を持つなら一意数群も合わせて更新し、持っていないなら一意数群を新規に求める処理を行う。

【手順 4】(数字の確定 3)

長さ1の区切り列では、その合計がそのマスの確定数となる。

*手順4から手順1に戻り、これを繰り返す。

2) 手順2を行うための除外処理

〔除外判定〕 空きマスを考える。ある数字がつぎの条件：

- ① マスを含むタテとヨコのスタート時の区切り列において、すでに決まっている確定数のどれかと一致する,
 - ② マスが一群マスで、その一意数群のどれとも一致しない,
 - ③ マスを含む区切り列が長さ2かつ合計が偶数で、合計の半分の値と一致する、
の一つをみれば、その除外数となる。
- さて、この除外判定を用いた手法をあげる。

【手法1】長さ2の区切り列の一方のマスを考える。数字 k がつぎの条件をみたすとき、その除外数となる。

「(合計 $-k$)として他方のマスに対応する数字 m を作る。

- ① 数字 m が、他方のマスで除外判定の条件をみたす。
- ② 他方のマスがさらに別の長さ2の区切り列に含まれ、同様にして作った数字 n が同様に対象とするマスで除外判定の条件をみたす。
- ③ 長さ2の区切り列が続く限り同様の処理を行い、ある段階で除外判定の条件がみたされる。」

3) 上記の手順や除外条件だけではほとんどの問題が解けないので、つぎを追加する。

【除外判定の更新】除外判定につぎの条件を追加する。

- ④ マスを含む区切り列において、そのマスを除いた残りの空きマス列が取り得る合計の最大値 MX と最小値 MN に対し、(合計 $-MX$)未満か(合計 $-MN$)を超える。

(例えば、区切り列の長さが3で合計が22なら、各マスに入り得る数字は5以上となる。何故なら、それを除いたマス列の数は2でその取り得る最大合計は17だから、4以下の数字を入れたら、残りのマスにどんな数字を入れようとその合計は21以下になる。)

(注意2) 可能数リストの見地からいうと、(合計 $-MX$)以上かつ(合計 $-MN$)以下の範囲内の数字がその可能数リストになる。

【手順5】(可能性リストの追加)

区切り列の長さが2で一方が一群マスなら、一群マスでない他方のマスの可能性リストは、その合計から一方の一意数群を引いた数字からなる集まりとして求める。

【手順6】手順1のダブル数字、及び手順2①の除外した残り数字が2つの場合、両方の数字の処理を平行して進めていく。特に、ある時点で一方がダメになれば(例えば、区切り列に同じ数字が重複して入る)、他方が確定する。

2.2 解を求めるためのより高度な手法

1) 対確定群の定義と分離

手順6で扱った可能数リストが2つのものをより組織的に扱う。

(定義 2.3) 区切り列の中の2つのマスにおいて、それらの可能数リストが2つの同じ数字からなるとき、この2つのマスに対確定群、及び2つの同じ数字を対確定数と呼ぶ。

【手順7】(対確定群の分離)

長さ3以上の区切り列が対確定群を持つなら、その対確定群をそこから分離し、取り去った残りのマス列を新しい区切り列とする。その合計は元の合計から対確定数の和を引いたものとなる。また、元の区切り列が一意数群も持つなら、新しい区切り列の一意数群は対確定群を取り去ったもの、持っていないなら一意数群を新規に求める処理を行う。

2) α 最大数と α 最小数

右の合計32の区切り列の中には、数字8が必ず入る。

	9			
--	---	--	--	--

もし8が入らないなら、4つの空きマスの取り得る最大

の合計は22 ($7 + 6 + 5 + 4$) で、 $22 + 9 < 32$ だから。

(定義 2.4) ある区切り列において、そのマスのどれかに入ることが保証される数字を入力数と呼ぶ。ある入力数に関し、入る可能性があるマス全体を可能マスリストと呼ぶ。

(定義 2.5) ある区切り列に入り得る最大数 [及び最小数] が入力数であるとき、 α 最大数 [及び α 最小数] と呼ばれる。

【手順8】(α 最大数・ α 最小数の確定処理)

α 最大数 [及び α 最小数] を持つ区切り列のあるマスにおいて、 α 最大数 [及び α 最小数] が他のどのマスでも除外数となるなら、その確定数となる。

3) 一意数群の制約撤去

一意数群は1つの可能数リストより、手順2①の一意数群を可能数リストに置き換える。

【手順9】ある空きマスにおいて、その可能数リストから除外数を取り去った結果として一つの数字が残ったら、残った数字はその確定数となる。

4) 限定された使用数での高度処理

確定数が決まるたびに、対象となる区切り列はせばまり、また各マスに入り得る数字 (すなわち、使用できる数字) は限定される。従って、一意数群や区切り列に関する取り得る合計の最大値や最小値は、その点を考慮して求めなければならない。

【手順10】ある区切り列において、各マスの可能数リストから重複しない数字のすべての選択による和を取るとき、与えられた合計数と一致するものが一組しかないなら、区切り列のすべてのマスはこの組の数字で確定する。

5) 矛盾処理

(定義 2.6) 2つの数字からなる可能数リストを持つマス [及び2つのマスからなる可能マスリストを持つ数字] において、可能数リストの一方を確定数 [及び可能マスリストの一方でその数字

が確定する]と仮定し、その条件追加によりさらなる数字の確定を行っていく。その経過中でダメな結果が起こったら（例えば、区切り列に同じ数字が重複する）、スタートしたマスには可能数リストの他方の数字[及び可能マスリストの他方のマスにその数字]が確定する。この原理を用いた処理法を矛盾処理と呼ぶ。

可能数リスト[及び可能マスリスト]の取り方としてつぎのものが考えられる。

- ① 長さ2の区切り列の一意数群 ② 対確定群
- ③ 手順7①で求まった2つの数字からなる可能数リスト
- ④ 長さ2の一意数群を持たない区切り列で手順9による可能数リスト
- ⑤ 手順7②で除外された結果残ったマスが2つの場合
- ⑥ 手順9で除外された結果残ったマスが2つの場合

(注意3) 手順6の処理は、①及び④を可能数リストとする矛盾処理に含まれる。

3. 難易度の基準化

3.1 難易についての段階とそれを決める基準

段階としてはレベル1～レベル5までの5段階とする。

さて、難易を決める基準であるが、すでに扱ってきたパズルと同様、カツクロにおいても問題の本来持っている難しさや厄介さに加えて、解くときの手間とか面倒くささを考慮すべきであろう。

そこで、「2. カツクロの解き方」で示した一意数群を核にした基本的解法と高度な手法に応じた基準を考える。そのため、難しさに準拠する高度別に分けられた手法をいくつか用意する。

しかし、人間がやる場合で特に注意すべきは、ケースが2つの矛盾処理を結構早い段階で上手に使う点である。この点をレベルの基準にも含ませたい。

3.2 具体的手法のリストとそのレベル

手法の名称	手法の内容	難易レベル
UNIBUNKAI	無条件による一意数群を求める処理	基本処理
BUNKAIITI	手順1による確定数を見つける処理	〃
NOKOIMASU	手順4による確定数を見つける処理	〃
KOSAPIECE	手法1による手順2①の処理（長さ3まで）	レベル1
KOSAPIECE	手法1による手順2①の処理	レベル2
UGSUKETU	手順2②による確定数を見つける処理（長さ3まで）	〃
MUJYUNSYORI	可能数リストを①とする矛盾処理	〃
UNIBUNKAI	使用数を限定した一意数群を求める処理	レベル3
UGSUKETU	手順2②による確定数を見つける処理	〃
KUMI2SYORI	対確定群に関する処理	〃
PLEN2SYORI	手順10による確定数を見つける処理（長さ2に限定）	〃
PLEN3SYORI	手順9による可能数リスト1を見つける処理（長さ3）	〃
MUJYUNSYORI	可能数リストを②と③とする矛盾処理	〃
MAXMIN	使用数を限定した最大値と最小値を求める処理	レベル4
SAIDAIISO	手順8による確定数を見つける処理	〃
HIPLANSYORI	手順9による可能数リスト1を見つける処理（長さ4～5）	〃
MUJYUNSYORI	可能数リストを④とする矛盾処理	〃
SUPSUMHAN	手順10による確定数を見つける処理	レベル5
HISAIIDAIISO	使用数限定と手順8による確定数を見つける処理	〃
MUJYUNSYORI	可能数リストを⑤とする矛盾処理	〃

3.3 解くときの手間や面倒くさを反映させる基準

これは矛盾処理をどれだけ行うかによって決まる。そこで、矛盾処理を行う対象の大きさとして完成比率－確定数の全体に対する割合－を考え、つぎのように決める。

- ① 完成比率が50%未満で解決したレベルが2のとき、難易度をレベル3とする。
- ② 完成比率が50%未満で解決したレベルが3のとき、難易度をレベル4とする。
- ③ レベル4や5でも矛盾処理を行うが、難易度は完成比率には無関係とする。

4. レベルを計算するプログラム

```
rem ** KATUKURO **
' -- hairetu_sengen --
dim K(51,51),B(1,400,9),PN(50,50,1), KS(1,400,8),KP(1,400,8)
dim HK(51,51),HB(1,400,9),KH(400,3)
dim LWA(1,400,3),LIT(1,400,2),LPO(1,400,9),PK(1,400)
dim HW(1,400,2),HP(1,400,9),HPK(1,400), HKS(1,400,8),HKP(1,400,8)
dim KV(300,3), MX(9,9),JMX(9,9),CK(9,3),KM(9,3)
```

```

dim MAX(8),MIN(8),R(10),RN(10), HLS(3,9),MXNS(9)
dim HA(9),JY(9),N(9),M(9),O(9),SDK(2),TP(1)
' -- data_yomi --
gosub *DATAYOMI
' -- syoki_settei --
TP(0)=YPN:TP(1)=TPN
for I=1 to 8: read MIN(I),MAX(I): next I
data 1,9, 3,17, 6,24, 10,30, 15,35, 21,39, 28, 42, 36,44
gosub *KTISU: MASUTOTAL=KZRN
FKAN=0
' --- START ---
LEVEL=1
while LEVEL<=5
' -- start_settei --
for TY=0 to 1: for PN=1 to TP(TY)
LWA(TY,PN,1)=LWA(TY,PN,0): LWA(TY,PN,2)=LWA(TY,PN,3)
PK(TY,PN)=0
for I=0 to 9: B(TY,PN,I)=0: next I
for I=0 to 8: KS(TY,PN,I)=0:KP(TY,PN,I)=0: next I
next PN,TY
for T=1 to GN: for Y=1 to RN
if K(T,Y)>0 then K(T,Y)=0
next Y,T
'-- PN,LPO_settei --
gosub *LPNKETU
gosub *LPOKETU
' -- first_clue --
FCLUE=1
for TY=0 to 1
for PN=1 to TP(TY): gosub *UNIBUNKAI: next PN
next TY
FCLUE=0
' -- Kai_sagasi --
MUHAN=0: FHIRITU=0:KANHI=100
FSYORI2=1
while FSYORI2=1
FSYORI2=0: FMU=0
' -- syori1 --
FSYORI1=1
while FSYORI1=1
FSYORI1=0: ZOKA=1
' -- kihon_syori --
while ZOKA=1
ZOKA=0
' nokori_1-masu
gosub *NOKO1MASU
' mujyun?
if FMU=1 then goto *MUSYORI

```

```

' D-bunkai_itti
  if ZOKA=0 then gosub *BUNKAIITI
wend
' -- level_syori --
gosub *KOSAPIECE
if FMU=1 then goto *MUSYORI
,

if (LEVEL>1)and(ZOKA=0) then gosub *UGSUKETU
if LEVEL>2 then
  if (ZOKA=0)and(MUHAN=0) then gosub *KUMI2SYORI
  if ZOKA=0 then gosub *PLEN2SYORI
  if ZOKA=0 then gosub *PLEN3SYORI
end if
if LEVEL>3 then
  if (LEVEL=4)and(ZOKA=0) then gosub *SAIDAISYO
  if (ZOKA=0)and(MUHAN=0) then gosub *HIPLENSYORI
end if
if (LEVEL=5)and(MUHAN=0) then
  if ZOKA=0 then gosub *SUPSUMHAN
  if ZOKA=0 then gosub *HISAIDAISYO
end if
' tuzukeru?
if ZOKA=1 then FSYORI1=1
wend
' -- kansei_hantei(hiritu) --
gosub *KTISU
if KAN=MASUTOTAL then FKAN=1: goto *KANS
,

if (LEVEL>1)and(FHIRITU=0) then
  KANHI=int(KAN/MASUTOTAL*100+.5)
  FHIRITU=1
end if
' -- mujyun_syori --
*MUSYORI
if LEVEL>1 then
  gosub *MUJYUNSYORI
  if FMSYOEND=0 then FSYORI2=1
end if
wend
' -- level_up --
LEVEL=LEVEL+1
wend
' kai_hyouji
*KANS
if FKAN=1 then
  ' - level_kettei -
  if ((LEVEL=2)or(LEVEL=3))and(KANHI<50) then LEVEL=LEVEL+1
  locate 10,21:print "LEVEL=";LEVEL

```

```

' - kai_hyoji -
gosub *KAIHYOJI
else
  locate 10,21:print "TOKENAI !!"
end if
end
:::
::: ' Sub Rutin :::
*KTISU
  KZRN=0:KAN=0
  for T=1 to GN:for Y=1 to RN
    if K(T,Y)=0 then KZRN=KZRN+1
    if K(T,Y)>0 then KAN=KAN+1
  next Y,T
  return
::
*DATAYOMI
  FILE$="a:ktprob.dat"
  open FILE$ for input as #1
  '
  input #1,GN,RN
  ' syoki_settei
  for I=1 to GN: K(I,0)=-1:K(I,RN+1)=-1: next I
  for J=1 to RN: K(0,J)=-1:K(GN+1,J)=-1: next J
  ' K
  for T=1 to GN
    input #1,JN
    SJ=0
    for P=1 to JN
      input #1,I,J
      for Y=SJ+1 to SJ+J
        if I=0 then K(T,Y)=-1 else K(T,Y)=0
      next Y
      SJ=SJ+J
    next P
  next T
  ' LIT,LWA_1
  YPN=0
  for I=1 to GN:for J=1 to RN+1
    if (K(I,J)=0)and(K(I,J-1)=-1) then YPN=YPN+1:NS=J
    if (K(I,J)=-1)and(K(I,J-1)=0) then
      LIT(0,YPN,1)=I : LIT(0,YPN,2)=NS
      LWA(0,YPN,0)=J-NS
    end if
  next J,I
  TPN=0
  for J=1 to RN:for I=1 to GN+1
    if (K(I,J)=0)and(K(I-1,J)=-1) then TPN=TPN+1:NS=I

```

```

    if (K(I,J)=-1)and(K(I-1,J)=0) then
      LIT(1,TPN,1)=NS : LIT(1,TPN,2)=J
      LWA(1,TPN,0)=I-NS
    end if
  next I,J
' LWA_2
for I=1 to YPN
  input #1,LW: LWA(0,I,3)=LW
next I
for I=1 to TPN
  input #1,LW: LWA(1,I,3)=LW
next I
return
::
*LPNKETU
for TY=0 to 1
  for T=1 to GN:for Y=1 to RN
    if K(T,Y)<>-1 then
      for Q=1 to TP(TY)
        if TY=0 then
          if (LIT(TY,Q,1)=T)and(LIT(TY,Q,2)<=Y) then PN(T,Y,TY)=Q
        else
          if (LIT(TY,Q,1)<=T)and(LIT(TY,Q,2)=Y) then PN(T,Y,TY)=Q
        end if
      next Q
    end if
  next Y,T
next TY
return
::
*LPOKETU
for TY=0 to 1
  for PN=1 to TP(TY): for I9=1 to LWA(TY,PN,1)
    LPO(TY,PN,I9)=LIT(TY,PN,2-TY)+(I9-1)
  next I9,PN
next TY
return
::
*KVSET
' -- par. T,Y,SU --
ZOKA=1
KVN=KVN+1: KV(KVN,1)=T:KV(KVN,2)=Y:KV(KVN,3)=SU
return
::
*UNIBUNKAI
' - par. LEVEL, TY,PN -
LWA1=LWA(TY,PN,1):LWA2=LWA(TY,PN,2)
if (B(TY,PN,0)=0)and((FCLUE=1)or(LEVEL<3)) then

```

```

for U9=1 to 9: N(U9)=0: next U9
else
  MTY=TY:MPN=PN:gosub *RIYOSU
end if
'

gosub *LSEN
if HAN=1 then
  ZOKA=1: B(TY,PN,0)=LWA1
  for U9=1 to LWA1: B(TY,PN,U9)=HA(U9): next U9
end if
return
::
*RIYOSU
' -- par. MTY,MPN --
for IR=1 to 9: N(IR)=0: next IR
R1=LIT(MTY,MPN,1+MTY):R2=LIT(MTY,MPN,2-MTY)
for IR=R2 to R2+LWA(MTY,MPN,0)-1
  if MTY=0 then KQ=K(R1,IR) else KQ=K(IR,R1)
  if KQ>0 then N(KQ)=1
next IR
' tui_bubun
for IR=1 to KS(MTY,MPN,0):for JR=1 to 2
  N(KS(MTY,MPN,2*(IR-1)+JR))=1
next JR,IR
return
::
*SETHAN
' -- par. SU, MTY,MPN, MT,MY, N --
if KS(MTY,MPN,0)=0 then BTY=1 else gosub *BSYOZOKU
'

if BTY=2 then
  gosub *KMNUM
  if KMN=0 then FSH=0:return
  for ISE=1 to 2
    if SU=KS(MTY,MPN,2*(KMN-1)+ISE) then FSH=0: return
  next ISE
  FSH=1
else
  ' BTY=1
  if B(MTY,MPN,0)>0 then
    for ISE=1 to B(MTY,MPN,0)
      if SU=B(MTY,MPN,ISE) then FSH=0: return
    next ISE
    FSH=1
  else
    FSH=1
    STW1=LWA(MTY,MPN,1):STW2=LWA(MTY,MPN,2)
    if N(SU)=1 then return

```

```

LW1=STW1:gosub *MAXMIN
if (MAX+SU<STW2)or(MIN+SU>STW2) then return
if STW1=2 then
  if ((STW2 mod 2)=0)and(STW2/2=SU) then return
  if N(STW2-SU)=1 then return
end if
,

FSH=0
end if
end if
return
::
*MAXMIN
' -- par. LW1, N --
if LEVEL<=3 then
  MAX=MAX(LW1-1):MIN=MIN(LW1-1)
else
  K=0:FK=0:MAX=0
  while FK<LW1-1
    if N(9-K)=0 then FK=FK+1:MAX=MAX+9-K
    K=K+1
  wend
  K=0:FK=0:MIN=0
  while FK<LW1-1
    if N(1+K)=0 then FK=FK+1:MIN=MIN+1+K
    K=K+1
  wend
end if
return
::
*BSYOZOKU
' -- par. MTY,MPN, MT,MY --
BTY=1
for IB=1 to LWA(MTY,MPN,1)
  G1=LIT(MTY,MPN,1+MTY):G2=LPO(MTY,MPN,IB)
  if MTY=0 then TB=G1:YB=G2 else TB=G2:YB=G1
  if (MT=TB)and(MY=YB) then return
next IB
BTY=2
return
::
*KMNUM
' -- par. MTY,MPN, MT,MY --
KMN=0
for IK=1 to KS(MTY,MPN,0):for JK=1 to 2
  G1=LIT(MTY,MPN,1+MTY):G2=KP(MTY,MPN,2*(IK-1)+JK)
  if MTY=0 then TB=G1:YB=G2 else TB=G2:YB=G1
  if (MT=TB)and(MY=YB) then KMN=IK:KMJ=JK: return

```

```

next JK,IK
return
::
*BUNKAIITI
KVN=0
for PN=1 to TP(0)
if (PK(0,PN)=0)and(B(0,PN,0)>1) then
for PO=1 to LWA(0,PN,1)
T=LIT(0,PN,1):Y=LPO(0,PN,PO)
'tate_bunkai
KPN=PN(T,Y,1)
if B(1,KPN,0)>1 then
for I9=1 to 9: N(I9)=0: next I9
for I9=1 to B(0,PN,0): N(B(0,PN,I9))=1: next I9
for I9=1 to B(1,KPN,0): N(B(1,KPN,I9))=N(B(1,KPN,I9))+1: next I9
ITIN=0
for I9=1 to 9
if N(I9)=2 then ITIN=ITIN+1:M(ITIN)=I9
next I9
if ITIN=1 then SU=M(1):gosub *KVSET
end if
next PO
end if
next PN
' new-LPO,LWA,uni-bunkai
if KVN>0 then gosub *NEWSETEI
return
::
*NEWSETEI
IN=1
while IN<=KVN
T=KV(IN,1):Y=KV(IN,2): K(T,Y)=KV(IN,3)
for TY=0 to 1
PN=PN(T,Y,TY): MTY=TY:MPN=PN:MT=T:MY=Y
gosub *BSYOZOKU
if BTY=1 then
gosub *NEWLPOWA
if LWA(TY,PN,1)>1 then gosub *UNIBUNKAI
if LWA(TY,PN,1)=0 then
if KS(TY,PN,0)=0 then PK(TY,PN)=1 else gosub *TUIBTYKA
end if
else
' BTY=2
gosub *KMNUM
if KMN>0 then
' new-K(tui-su)_tuika
G1=LIT(TY,PN,1+TY):G2=KP(TY,PN,2*(KMN-1)+(3-KMJ))
if TY=0 then TG=G1:YG=G2 else TG=G2:YG=G1

```

```

for J9=1 to 2
  if K(T,Y)=KS(TY,PN,2*(KMN-1)+J9) then SU=KS(TY,PN,2*(KMN-1)+3-J9)
next J9
KVN=KVN+1
KV(KVN,1)=TG:KV(KVN,2)=YG:KV(KVN,3)=SU
' KMN-tui_tori
for ISE=KMN to KS(TY,PN,0)-1:for JSE=1 to 2
  KP(TY,PN,2*(ISE-1)+JSE)=KP(TY,PN,2*ISE+JSE)
  KS(TY,PN,2*(ISE-1)+JSE)=KS(TY,PN,2*ISE+JSE)
next JSE,ISE
KS(TY,PN,0)=KS(TY,PN,0)-1
end if
end if
next TY
IN=IN+1
wend
return
::
*NEWLPOWA
LPOW=0:WA=LWA(TY,PN,2)
for N9=1 to LWA(TY,PN,1)
  G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,N9)
  if TY=0 then TG=G1:YG=G2 else TG=G2:YG=G1
  ' new LPO_sakusei
  if K(TG,YG)=0 then
    LPOW=LPOW+1:LPO(TY,PN,LPOW)=G2
  else
    WA=WA-K(TG,YG)
  end if
next N9
' new LWA_sakusei
LWA(TY,PN,1)=LPOW:LWA(TY,PN,2)=WA
return
::
*TUIBTYKA
' -- par. TY,PN --
KMN=KS(TY,PN,0)
LWA(TY,PN,1)=2:B(TY,PN,0)=2
for I9=1 to 2
  LPO(TY,PN,I9)=KP(TY,PN,2*(KMN-1)+I9)
  B(TY,PN,I9)=KS(TY,PN,2*(KMN-1)+I9)
next I9
LWA(TY,PN,2)=B(TY,PN,1)+B(TY,PN,2)
KS(TY,PN,0)=KMN-1
return
::
*NOKOIMASU
KVN=0

```

```

for TY=0 to 1
for PN=1 to TP(TY)
if (PK(TY,PN)=0)and(LWA(TY,PN,1)=1) then
WA=LWA(TY,PN,2)
G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,1)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
' mujiyun?
if (WA<1)or(WA>9) then FMU=1: return
MTY=TY:MPN=PN:gosub *RIYOSU
if N(WA)=1 then FMU=1: return
' kosa_piece
MTY=1-TY:MPN=PN(T,Y,MTY):gosub *RIYOSU
SU=WA: MT=T:MY=Y:gosub *SETHAN
if FSH=1 then FMU=1: return
' kakutei
gosub *KVSET: goto *NK1KEND
end if
next PN
next TY
' new-LPO,LWA,uni-bunkai
*NK1KEND
if KVN>0 then gosub *NEWSETEI
return
::
*KOSAPIECE
KVN=0
for TY=0 to 1
for PN=1 to TP(TY)
WA1=LWA(TY,PN,1)
if (((LEVEL=1)and(WA1<=3))or(LEVEL>1))and(PK(TY,PN)=0)and(B(TY,PN,0)>1) then
for I=0 to WA1: SL(I)=B(TY,PN,I): next I
for PO=1 to WA1
G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
gosub *TYSETSU
if KSN=0 then FMU=1: return
if KSN=1 then SU=M(1):gosub *KVSET: goto *KPEND
next PO
end if
next PN
next TY
' new-LPO,LWA,uni-bunkai
*KPEND
if KVN>0 then gosub *NEWSETEI
return
::
*UGSUKETU
KVN=0

```

```

for TY=0 to 1
  for PN=1 to TP(TY)
    WA1=LWA(TY,PN,1)
    if (((LEVEL=2)and(WA1<=3))or(LEVEL>2))and(PK(TY,PN)=0)and(B(TY,PN,0)>1) then
      for I=1 to WA1
        SU=B(TY,PN,I): GOSUB *MASSETSU
        IF KSN=1 THEN T=ML(1,1):Y=ML(1,2):GOSUB *KVSET: GOTO *USEND
      next I
    end if
  next PN
next TY
' new-LPO,LWA,uni-bunkai
*USEND
if KVN>0 then gosub *NEWSETEI
return
::
*TYSETSU
' -- par. TY,PN,T,Y, SL --
KSN=0
for IT=1 to SL(0)
  SU=SL(IT):gosub *KOSASETHAN
  if FSH=0 then KSN=KSN+1:M(KSN)=SU
next IT
return
::
*MASSETSU
' -- par. SU, TY,PN --
KSN=0
for PO=1 to LWA(TY,PN,1)
  G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
  if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
  gosub *KOSASETHAN
  if FSH=0 then KSN=KSN+1:ML(KSN,1)=T:ML(KSN,2)=Y
next PO
return
::
*KOSASETHAN
' -- par. SU, TY,PN,T,Y --
' kosa_piece
MTY=1-TY:MPN=PN(T,Y,MTY):gosub *RIYOSU
MT=T:MY=Y:gosub *SETHAN
' doble_hantei
GSU=SU
LOOPN=1: gosub *BSYOZOKU
while (FSH=0)and(LWA(MTY,MPN,1)=2)and(BTY=1)and(LOOPN<=3)
  GTY=MTY:GPN=MPN:GTI=MT:GYI=MY
  for I9=1 to 2
    G3=LIT(GTY,GPN,1+GTY):G4=LPO(GTY,GPN,I9)

```

```

if GTY=0 then TK=G3:YK=G4 else TK=G4:YK=G3
if (GTI<>TK)or(GY1<>YK) then
' kosa_piece
MTY=1-GTY:MPN=PN(TK,YK,MTY):gosub *RIYOSU
MT=TK:MY=YK
SU=LWA(GTY,GPN,2)-SU:gosub *SETHAN
end if
next I9
' next-kosa_piece
gosub *BSYOZOKU: LOOPN=LOOPN+1
wend
SU=GSU
return
::
*PLEN2SYORI
KVN=0
for TY=0 to 1: for PN=1 to TP(TY)
WA1=LWA(TY,PN,1):WA2=LWA(TY,PN,2)
if (PK(TY,PN)=0)and(B(TY,PN,0)=0)and(WA1=2) then
SAKTY=1:gosub *JMXMAKE
'

MNTY=1:LWA1=2:LWA2=WA2:gosub *HLSSEN
' case_1
if HAN=1 then
G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,1)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
SU=HLS(1,1): gosub *KVSET
goto *PL2END
end if
' case_2
if (HAN=2)and(HLS(1,1)+HLS(2,1)=WA2) then
ZOKA=1: B(TY,PN,0)=2
for U9=1 to 2: B(TY,PN,U9)=HLS(1,U9): next U9
goto *PL2END
end if
end if
next PN,TY
' new-LPO,LWA,uni-bunkai
*PL2END
if KVN>0 then gosub *NEWSETEI
return
::
*JMXMAKE
' -- par. SAKTY,(JSU),TY,PN --
gosub *KANOLIST
for I9=0 to KN(0): SL(I9)=KN(I9): next I9
' JMX_make
for PO=1 to LWA(TY,PN,1)

```

```

G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
gosub *TYSETSU
JMX(PO,0)=KSN
for I9=1 to KSN: JMX(PO,I9)=M(I9): next I9
next PO
return
::
*KANOLIST
' -- par. MNTY,(JSU),TY,PN --
if B(TY,PN,0)>0 then
  if SAKTY=1 then
    for I9=0 to B(TY,PN,0): KN(I9)=B(TY,PN,I9): next I9
  else
    KSN=0
    for I9=0 to B(TY,PN,0)
      if JSU<>B(TY,PN,I9) then KSN=KSN+1: KN(KSN)=B(TY,PN,I9)
    next I9
    KN(0)=KSN
  end if
else
  KW1=LWA(TY,PN,1):KW2=LWA(TY,PN,2)
  MTY=TY:MPN=PN: gosub *RIYOSU
  LW1=KW1:gosub *MAXMIN
  for I9=1 to 9
    IF (MAX+I9<KW2)or(MIN+I9>KW2) then N(I9)=1
  next I9
  if (KW1=2)and((KW2 MOD 2)=0) then N(KW2/2)=1
  if SAKTY=2 then N(JSU)=1
  ' list_sakusei
  KSN=0
  for I9=1 to 9
    if N(I9)=0 then KSN=KSN+1:KN(KSN)=I9
  next I9
  KN(0)=KSN
end if
return
::
*KUMI2SYORI
KVN=0
for TY=0 to 1
  for PN=1 to TP(TY)
    if (PK(TY,PN)=0)and(LWA(TY,PN,1)>2) then
      gosub *KM2LIST
      ' double_syori
      if KMN>0 then ZOKA=1:gosub *SKSYO: goto *KMSEND
    end if
  next PN

```

```

next TY
*KMSEND
' new_settei
if KVN>0 then gosub *NEWSETEI
return
::
*SKSYO
WA1=LWA(TY,PN,1):WA2=LWA(TY,PN,2)
' new_LPOWA
for I=1 to WA1: N(I)=0: next I
NON=0:NOS=WA2
for I=1 to KMN:for J=1 to 2
  N(CK(KM(I,J),0))=1
  NOS=NOS-CK(KM(I,1),J)
next J,I
for I=1 to WA1
  if N(I)=0 then NON=NON+1:M(NON)=I
next I
LWA(TY,PN,1)=NON:LWA(TY,PN,2)=NOS
for I=1 to NON: LPO(TY,PN,I)=LPO(TY,PN,M(I)): next I
'

if NON=0 then gosub *TUIBTYKA: return
' type_1(NON=1)
if NON=1 then
  G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,M(1))
  if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
  KVN=1:KV(1,1)=T:KV(1,2)=Y:KV(1,3)=NOS: return
else 'type_2(NON>1)
  ' kari_oki
  HSU=0
  for I=1 to KMN:for J=1 to 2
    G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,CK(KM(I,J),0))
    if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
    K(T,Y)=CK(KM(I,1),J):HSU=HSU+1:HPK(0,HSU)=T:HPK(1,HSU)=Y
  next J,I
  ' uni_bunkai
  gosub *UNIBUNKAI
  ' modosi
  for I=1 to HSU: K(HPK(0,I),HPK(1,I))=0: next I
end if
return
::
*KM2LIST
' -- par. TY,PN --
GOSUB *KSU2LIST
' double?
KMN=0
if CKN>=2 then

```

```

for I8=1 to CKN-1:for I9=I8+1 to CKN
  if (CK(I8,1)=CK(I9,1))and(CK(I8,2)=CK(I9,2)) then
    KMN=KMN+1
    KM(KMN,1)=I8:KM(KMN,2)=I9
  end if
next I9,I8
' list_sakusei
if KMN>0 then
  ' KB,KPO_tuika
  KS=KS(TY,PN,0)
  KS(TY,PN,0)=KS+KMN
  for I=1 to KMN:for J=1 to 2
    KS(TY,PN,2*KS+2*(I-1)+J)=CK(KM(I,1),J)
    KP(TY,PN,2*KS+2*(I-1)+J)=LPO(TY,PN,CK(KM(I,J),0))
  next J,I
end if
end if
return
::
*KSU2LIST
' -- par. TY,PN --
SAKTY=1:gosub *JMXMAKE
CKN=0
for PO=1 to LWA(TY,PN,1)
  if JMX(PO,0)=2 then
    CKN=CKN+1: CK(CKN,0)=PO
    for I9=1 to 2: CK(CKN,I9)=JMX(PO,I9): next I9
  end if
next PO
return
::
*SAIDAISSYO
SDSTY=1:gosub *SDSSYORI
return
::
*SDSSYORI
' -- par. SDSTY --
KVN=0
for TY=0 to 1
  for PN=1 to TP(TY)
    if B(TY,PN,0)=0 then
      gosub *SDKOHO
      for I=1 to SDKSU
        SU=SDK(I):gosub *MASSETSU
        if KSN=1 then
          T=ML(1,1):Y=ML(1,2):gosub *KVSET: goto *SDSEND
        end if
      next I
    end if
  next PN
next TY

```

```

    end if
  next PN
next TY
' new-LPO,LWA,uni-bunkai
*SDSEND
if KVN>0 then gosub *NEWSETEI
return
::
*SDKOHO
' -- par. TY,PN,SDSTY --
SDKSU=0
SDW1=LWA(TY,PN,1):SDW2=LWA(TY,PN,2)
for MXTY=1 to 2
  MTY=TY:MPN=PN:gosub *RIYOSU
  if MXTY=1 then ST=9:SP=-1 else ST=1:SP=1
  for I9=ST to 10-ST step SP
    if N(I9)=0 then MXN=I9:goto *EMXN1
  next I9
  *EMXN1
  N(MXN)=1
  ' MAX,MIN_kettei
  if SDSTY=1 then
    KAWA=0:KSN=0
    for I9=ST to 10-ST step SP
      if N(I9)=0 then KSN=KSN+1:KAWA=KAWA+I9
      if KSN=SDW1 then *EMXN2
    next I9
    *EMXN2
    if MXTY=1 then MAX=KAWA else MIN=KAWA
  else
    ' sdsty=2
    SAKTY=2:JSU=MXN: gosub *JMXMAKE
    LWA1=SDW1:MNTY=2: gosub *HLSN
  end if
  ' hantei
  if ((MXTY=1)and(MAX<SDW2))or((MXTY=2)and(MIN>SDW2)) then
    SDKSU=SDKSU+1:SDK(SDKSU)=MXN
  end if
next MXTY
return
::
*PLEN3SYORI
MASUSU=3:gosub *HIPLN
return
::
*HIPLNSYORI
for MASUSU=4 to 6
  gosub *HIPLN

```

```

    if KVN>0 then return
next MASUSU
return
::
*HIPLN
' -- par. MASUSU --
KVN=0
for TY=0 to 1: for PN=1 to TP(TY)
    WA1=LWA(TY,PN,1):WA2=LWA(TY,PN,2)
    if (PK(TY,PN)=0)and(B(TY,PN,0)=0)and(WA1=MASUSU) then
        SAKTY=1:gosub *JMXMAKE
        ' case_1
        for PO=1 to WA1
            if JMX(PO,0)=1 then
                G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
                if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
                SU=JMX(PO,1):gosub *KVSET: goto *HPLEND
            end if
        next PO
        ' case_2
        if LEVEL>3 then
            for I2=0 to 1
                for PO=1 to WA1
                    if I2=0 then MXNS=1 else MXNS=9
                    for I9=1 to JMX(PO,0)
                        if I2=0 then
                            if MXNS<JMX(PO,I9) then MXNS=JMX(PO,I9)
                        else
                            if MXNS>JMX(PO,I9) then MXNS=JMX(PO,I9)
                        end if
                    next I9
                    MXNS(PO)=MXNS
                next PO
            ' wa
            SUM=0
            for I9=1 to WA1: SUM=SUM+MXNS(I9): next I9
            if SUM=WA2 then
                for PO=1 to WA1
                    G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
                    if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
                    SU=MXNS(PO): gosub *KVSET
                next PO
                goto *HPLEND
            end if
        next I2
    end if
end if
next PN, TY

```

```

' new_settei
*HPLEND
if KVN>0 then gosub *NEWSETEI
return
::
*SUPSUMHAN
for MASUSU=3 to 5
  gosub *SUPSUM
  if KVN>0 then return
next MASUSU
return
::
*SUPSUM
' -- par. MASUSU --
KVN=0
for TY=0 to 1: for PN=1 to TP(TY)
  WA1=LWA(TY,PN,1):WA2=LWA(TY,PN,2)
  if (PK(TY,PN)=0)and(B(TY,PN,0)=0)and(WA1=MASUSU) then
    SAKTY=1:gosub *JMXMAKE
  '
  MNTY=1:LWA1=WA1:LWA2=WA2:gosub *HLSN
  if HAN=1 then
    for PO=1 to WA1
      G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
      if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
      SU=HLS(1,PO): gosub *KVSET
    next PO
    goto *SSHEND
  end if
end if
next PN, TY
' new_settei
*SSHEND
if KVN>0 then gosub *NEWSETEI
return
::
*HISAIDAISSYO
SDSTY=2:gosub *SDSSYORI
return
::
*MUJYUNSYORI
FMSYOEND=0
'
if MUHAN=0 then
  MUHAN=1
  ' - koho_list -
  gosub *KOHOLIST
  ' - hikae_toru -

```

```

HTY=0:gosub *HIKAE
' - atai_set -
KVA=1: SETTI=KVA
else
' - hikaе_modosi -
HTY=1:gosub *HIKAE
' - mujyun_ari? -
if FMU=1 then
if (KVA mod 2)=0 then SETTI=KVA-1 else SETTI=KVA+1
MUHAN=0
else
KVA=KVA+1
if KVA>KHN then FMSYOEND=1: return
SETTI=KVA
end if
end if
' K-ti_settei
KVN=1
for I=1 to 3: KV(1,I)=KH(SETTI,I): next I
gosub *NEWSETEI
return
::
*KOHOLIST
KHN=0
' case_1
for TY=0 to 1: for PN=1 to TP(TY)
if (PK(TY,PN)=0)and(B(TY,PN,0)=2) then
G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,1)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
for I=1 to 2
KHN=KHN+1
KH(KHN,1)=T:KH(KHN,2)=Y:KH(KHN,3)=B(TY,PN,I)
next I
end if
next PN,TY
if LEVEL=2 then return
' case_2
for TY=0 to 1: for PN=1 to TP(TY)
if KS(TY,PN,0)>0 then
for I=1 to KS(TY,PN,0)
G1=LIT(TY,PN,1+TY):G2=KP(TY,PN,2*(I-1)+1)
if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
for J=1 to 2
KHN=KHN+1
KH(KHN,1)=T:KH(KHN,2)=Y:KH(KHN,3)=KS(TY,PN,2*(I-1)+J)
next J
next I
end if

```

```

next PN,TY
' case_3
for TY=0 to 1: for PN=1 to TP(TY)
  WA1=LWA(TY,PN,1)
  if B(TY,PN,0)>2 then
    SAKTY=1:gosub *JMXMAKE
    for PO=1 to LWA(TY,PN,1)
      G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,PO)
      if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
      if JMX(PO,0)=2 then
        for I=1 to 2
          KHN=KHN+1: KH(KHN,0)=0
          KH(KHN,1)=T:KH(KHN,2)=Y:KH(KHN,3)=JMX(PO,I)
        next I
      end if
    next PO
  end if
next PN,TY
' case_4
for TY=0 to 1: for PN=1 to TP(TY)
  WA1=LWA(TY,PN,1):WA2=LWA(TY,PN,2)
  if (PK(TY,PN)=0)and(B(TY,PN,0)=0)and(WA1=2) then
    SAKTY=1:gosub *JMXMAKE
    ,
    MNTY=1:LWA1=2:LWA2=WA2:gosub *HLSSEN
    if HAN=2 then
      G1=LIT(TY,PN,1+TY):G2=LPO(TY,PN,1)
      if TY=0 then T=G1:Y=G2 else T=G2:Y=G1
      ,
      for I=1 to 2
        KHN=KHN+1: KH(KHN,0)=0
        KH(KHN,1)=T:KH(KHN,2)=Y:KH(KHN,3)=HLS(I,1)
      next I
    end if
  end if
next PN,TY
if LEVEL=3 then return
' case_5
for TY=0 to 1: for PN=1 to TP(TY)
  WA1=LWA(TY,PN,1)
  if B(TY,PN,0)>2 then
    for I=1 to B(TY,PN,0)
      SU=B(TY,PN,I):gosub *MASSETSU
    if KSN=2 then
      for J=1 to 2
        KHN=KHN+1
        KH(KHN,1)=ML(J,1):KH(KHN,2)=ML(J,2):KH(KHN,3)=SU
      next J
    end if
  end if
next PN,TY

```

```

        end if
    next I
end if
next PN, TY
if LEVEL=4 then return
' case_6
for TY=0 to 1: for PN=1 to TP(TY)
    if (LWA(TY, PN, 1)>2)and(B(TY, PN, 0)=0) then
        gosub *SDKOHO
        for I=1 to SDKSU
            SU=SDK(I):gosub *MASSETSU
            if KSN=2 then
                for J=1 to 2
                    KHN=KHN+1
                    KH(KHN, 1)=ML(J, 1):KH(KHN, 2)=ML(J, 2):KH(KHN, 3)=SU
                next J
            end if
        next I
    end if
next PN, TY
return
::
*HIKAE
for T=1 to GN:for Y=1 to RN
    if HTY=0 then HK(T, Y)=K(T, Y) else K(T, Y)=HK(T, Y)
next Y, T
for TY=0 to 1:for PN=1 to TP(TY)
    for I=0 to 9
        if HTY=0 then HB(TY, PN, I)=B(TY, PN, I) else B(TY, PN, I)=HB(TY, PN, I)
    next I
    if HTY=0 then HPK(TY, PN)=PK(TY, PN) else PK(TY, PN)=HPK(TY, PN)
    for I=1 to 2
        if HTY=0 then HW(TY, PN, I)=LWA(TY, PN, I) else LWA(TY, PN, I)=HW(TY, PN, I)
    next I
    for I=1 to 9
        if HTY=0 then HP(TY, PN, I)=LPO(TY, PN, I) else LPO(TY, PN, I)=HP(TY, PN, I)
    next I
    for I=0 to 8
        if HTY=0 then HKP(TY, PN, I)=KP(TY, PN, I) else KP(TY, PN, I)=HKP(TY, PN, I)
        if HTY=0 then HKS(TY, PN, I)=KS(TY, PN, I) else KS(TY, PN, I)=HKS(TY, PN, I)
    next I
next PN, TY
return
::
*LSEN
' -- par. LWA1, LWA2, N --
for J9=1 to 10: RN(J9)=9-LWA1+J9: next J9
K=1:HAN=0

```

```

R(1)=LWA2-MAX(LWA1-1): if R(1)<1 then R(1)=1
while K>0
  if R(K)<=RN(K) then FOWARD=1 else FOWARD=0
  on FOWARD+1 gosub *BACK,*FOWARD
  if K>LWA1 then gosub *HANTEI:gosub *BACK
wend
return
:::
*FOWARD
K=K+1
R(K)=R(K-1)+1
return
::
*BACK
K=K-1
R(K)=R(K)+1
return
::
*HANTEI
TOTAL=0
for W9=1 to K-1
  if N(R(W9))=1 then return
  TOTAL=TOTAL+R(W9)
next W9
if TOTAL=LWA2 then
  HAN=HAN+1
  for W9=1 to K-1: HA(W9)=R(W9): next W9
end if
return
::
*HLSSEN
' -- par. MNTY,LWA1,LWA2, JMX --
if MNTY=1 then HAN=0 else MAX=0:MIN=50
for H1=1 to LWA1: RN(H1)=JMX(H1,0): next H1
K=1:R(1)=1
WHILE K>0
  if R(K)<=RN(K) then FOWARD=1 else FOWARD=0
  on FOWARD+1 gosub *HBACK,*HFOWARD
  if K>LWA1 then
    gosub *HHANTEI
    if (MNTY=1)and(HAN>2) then return
    gosub *HBACK
  end if
wend
return
::
*HFOWARD
K=K+1: R(K)=1

```

```

return
::
*HBACK
K=K-1: R(K)=R(K)+1
return
::
*HHANTEI
for H9=1 to 9: N(H9)=0: next H9
for H9=1 to K-1: N(JMX(H9,R(H9)))=1: next H9
FDN=0
for H9=1 to 9
  IF N(H9)=1 THEN FDN=FDN+1
next H9
if FDN=K-1 then
  TOTAL=0
  for H9=1 to K-1: TOTAL=TOTAL+JMX(H9,R(H9)): next H9
  '
  if MNTY=1 then
    if TOTAL=LWA2 then
      HAN=HAN+1
      for H9=1 to K-1: HLS(HAN,H9)=JMX(H9,R(H9)): next H9
    end if
  else
    if TOTAL>MAX then MAX=TOTAL
    if TOTAL<MIN then MIN=TOTAL
  end if
end if
return

```

5. 問題 [2] の結果とその考察

5.1 基準による難易度結果と問題に付随する難易度との適合性

問題集[2]の各冊とも、四角枠の大きさにより5つの部分 Part1 (11 × 9), Part2 (13 × 11), Part3 (13 × 9), Part4 (15 × 11), Part5 (19 × 13) に問題が分けられている。

さて、各問題は、難易度ではなく解くのに要する時間が付き、その時間順に並べられている。そこで、この時間をもとに5段階の難易度を設定する必要がある。設定のための一義的な方法がないので、実際の問題を解くのに関連させて設定せざるを得ない。

基準による難易度から得られる問題の解結果（題数）はつぎのようにまとめられる。

レベル	Part 1	Part 2	Part 3	Part 4	Part 5
1	13(5, 4, 4)	5(2, 2, 1)	6(2, 1, 3)	1(0, 1, 0)	0
2	23(8, 7, 8)	22(11, 6, 5)	24(13, 5, 6)	22(10, 7, 5)	6(5, 0, 1)
3	19(7, 5, 7)	26(7,11, 8)	25(4,11,10)	27(9, 8,10)	5(0, 3, 2)
4	9(1, 6, 2)	9(0, 2, 7)	9(2, 3, 4)	6(1, 3, 2)	11(1, 4, 6)
5	0	0	1(0, 1, 0)	1(0, 1, 0)	1(1, 0, 0)
合計	6 4 題	6 2 題	6 5 題	5 7 題	2 3 題

但し、13 (5,4,4) は1冊目,2冊目,3冊目の問題数がそれぞれ5題,4題,4題で合計13題となることを示す。

[1]では、各Partごとに解法時間（中級者のものを用いる）を上の比率に合わせた割合で比例配分し、それを5段階の難易度とした（冊ごとに解法時間と難易度がバラバラなので各冊ごとに決めた）。しかし、ここでは解法時間で決まる順序を重視し、順番に並んだものを上の比率に合わせた割合で比例配分し、それを5段階の難易度とする。例えば1冊目では、各問題に付随する難易度がつぎのように決まる。

レベル	1冊目の問題番号				
	Part 1	Part 2	Part 3	Part 4	Part 5
1	1, 3, 5, 7, 9	2, 4	42,44		
2	11,13,15,17,19 21,23,25	6, 8,10,12,14 16,18,20,22,24 26	46,50,48,52,54 56,58,60,62,64 66,68,72	43,47,49,55,45 51,53,57,59,65	83,84,85 86,87
3	27,29,31,33,35 41,37	28,30,32,34,36 38,40	70,74,76,78	61,67,63,69,73 71,75,77,79	
4	39		80,82	81	88
5					89

これと基準による難易度結果との適合性を調べ、まとめてつぎを得る。

問題に付随した難易度	基準によって決まる難易度					合計
	レベル1	レベル2	レベル3	レベル4	レベル5	
レベル1	15	3	7			25
〃 2	8	64	21	4		97
〃 3	2	27	57	16		102
〃 4		3	16	23	2	44
〃 5			1	1	1	3

この結果からつぎのことがいえる。

- 1) 適合率は、レベル1で60%、レベル2で66%、レベル3で56%、レベル4で52%、レベル5で33%である。問題数の少ないレベル5を除いて、全体では59.3%である。
- 2) 付随した難易度と基準によって決まる難易度との相関係数は、0.63である。

以上のことから、3節で定めた基準による難易度と人間の感じる難易度がかなり適合するのが相関係数の値からもわかる。しかし、もっと適合率を高めるためには、人間が解く方法である局所重視的な要素をもっと取り入れる必要があるだろう。

文 献

- [1] 佐藤金吾, パズル「スリザーリンク」の難易度について, 法政大学多摩研究報告, 18 (2003)
- [2] ペンシルパズル本『カックロ』1～3 (2002年4月 株式会社ニコリ)