

### 「イラストパズル」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)

23

(開始ページ / Start Page)

17

(終了ページ / End Page)

75

(発行年 / Year)

2008-03-30

(URL)

<https://doi.org/10.15002/00004287>



問題集 [2] ~ [4] を用いた。

③ 本分中のプログラムには、言語として「構造化 BASIC98」を用いた。

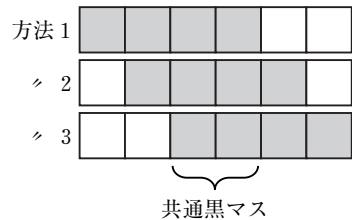
## 2. 「イラストパズル」の解き方

### 2.1 4つの基本処理

この種の数理パズルの解法に共通するのが、基本となるいくつかの処理の発見である。これらは、最初の手がかりを与えると同時に、新しい手法等で見つかった黒や白マスの情報を活かし、さらに多くのマス情報を生み出す核としての役割を果たす。

#### 1) 「ピース入れ」処理

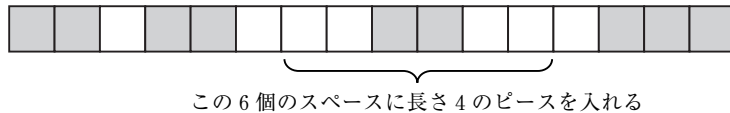
空きスペースにピースを入れる際に生じる黒マスを見つけ  
る処理。例えば、長さ6マスのスペースに長さ4マスのピー  
スを入れる時、右図のように3通りの方法があり、どれにも  
共通した黒マスが2個ある。



これらの共通マスはどんな入れ方をしようと必ず生ずる  
黒マスであり、このような黒マスを見つけるのが上記処理である。

複数のピースを入れる時も、この1個のピースを入れる場合に還元される。問題とするピースに対しては、それ以外のピースを、左と右から書かれた順番に、間に1個の白マスを挟んで隙間なく並べ、残ってできた空きスペースに問題のピースを入れればよい。この時、共通して生じる黒マスが求めるものである。

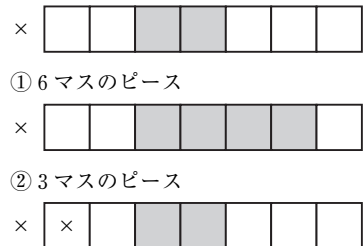
例えば、下図は長さ16マスのスペースに、長さ2,2,4,3マスの4つのピースを入れる場合で、長さ4マスのピースを問題としたものを表す。



#### 2) 「黒マス伸び、白マス埋め」処理

壁（外ワクか途中の白マス）に隣り合って長さの決まった  
ピースが入る状況で、その壁から離れて（接触の場合も含む）  
黒マスが存在する場合に生じる黒や白マスを見つける処理。

例えば、右図で壁から2マス分あけて2個の黒マス並びが  
ある時、①長さ6マスのピースの場合は右に2個の黒マスが



伸び、②長さ3マスのピースの場合は壁に隣接した1個のマスが白マスとなる。①の場合、存

在マスが長さ6マスのピースの一部となるので、1)と同様にピースの頭が壁につく場合等の可能性に共通して生じる黒マスを求めればよい。②の場合、もし壁に隣接したマスが黒マスになれば、この黒マスから生じるピースは決められた長さ3マスを超えてしまい、矛盾となる。

つぎの性質は上の説明から明らかに従う。

[命題1] ① 長さ $n$ マスの空きスペースに長さ $p$ マスのピースを入れる時、

- ・  $p > n$  なら、すべてのマスは白マスとなる。
- ・  $(p \leq n)$  かつ  $(2p > n)$  なら、中央の  $(2p - n)$  個のマスが黒マスになる。

② 長さ $n$ マスの空きスペースに、長さがそれぞれ  $p_1, p_2, \dots, p_k$  マスの  $k$  個のピースを入れる時、各長さ  $p_i$  マスのピースに対し、 $p_1 + p_2 + \dots + p_k = TP$  として、

$2p_i > n - (TP - p_i + k - 1)$  なら、 $(p_i - n + TP + k - 1)$  個のマスが黒マスになる。


③ 壁に隣り合って長さ $p$ マスのピースが入る状況で、壁からマス $s$ 個の空きスペースに続き $b$ 個の黒マス並びがある時、

- ・  $p > s + b$  なら、壁と反対方向に  $(p - s - b)$  個の黒マスが伸びる。
- ・  $(p + 1 > s)$  かつ  $(s + b > p)$  なら、壁から  $(s + b - p)$  個のマスが白マスとなる。

### 3) 「長さ1マスのピース並び」処理

壁に隣り合って長さ1のピース並びが入る状況という、2)の特別な場合を考える。

例えば、右図で壁から4マス分あけて1個の黒マスがある ① 1マス並びが2つ

時、①は長さ1の並びが2の場合で、黒マスの左隣りは白マス × 

②は長さ1の並びが3の場合で、黒マスの左右の ② 1マス並びが3つ

隣りマスは共に白マスとなる。 × 

[命題2] 長さ1マスのピース並びを $n$ とし、壁からマス $s$ 個

の空きスペースに続き $b$ 個の黒マスブロックがあるとする。

- ・  $(2n > s - 1)$  なら、黒ブロックの壁と向かい合う側の接触マスは白マスとなる。
- ・  $(b = 1)$  かつ  $(2n > s)$  なら、黒マスの左右の接触マスは共に白マスとなる。
- ・  $(b > 1)$  かつ  $(2n = s)$  なら、壁と黒ブロックの間に長さ1マスのピース $n$ 個が隙間なく入る。

(証明) 長さ1マスの $n$ 個のピースが入るためには、マス $(2n - 1)$ 個分のスペースが必要であることに注意する。すると、例えば、最初の主張は、もし該当するマスを黒マスとすれば、長さ1マスのピースとの間に仕切り白マスが1個は必要なので、残り空きスペースは $(s - 2)$ マスに減少する。しかし、これは仮定の条件に矛盾するので、その白マスが結論される。

### 4) 「空きマス1個で並ぶ黒マスブロックの分離」処理

黒マスブロックの2つが間に空きマス1個をおいて並んでいる。もしこの空きマスを黒マスとすれば、誕生した拡大黒マスブロックをその一部とするピースが存在しえない時、空きマス

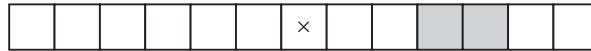
は白マスと結論される。この白マスによる分離処理をいう。

例えば、拡大ブロックの長さがピース並びの最大値を超える時は、この条件が満たされる。問題はその他の役立つ条件を求めることであり、以下の節の主要課題となる。

## 2.2 「ワク決め」及びピースの開始位置と終了位置の限度値

解処理がスタートした時点では、各列とも大きな一つの空きスペースである。しかし処理が進むにつれ、間に黒や白マスが生じてきて、解状況が変化してくる。

例えば、長さ13マスのスペースに長さ3,4マスの2つのピースを入れる時、「ピース入れ」処理から共通マスは生じない。しかし、下図のように真中のマスが白になり、2つのスペースに分かれると、長さ4マスのピースから2個の共通マスが生じる。



この黑白情報に応じてさまざまな解手法を生み出すために、必要な事項をあげる。

(定義1) 黒 [白] マス並びを黒 [白] ブロック、そのマスの数を長さという。

黒ブロックで、両端が白マスで閉じられたものを確定ブロック、そうでないものを未確定ブロックという。

マスのタテ [ヨコ] 位置とは、属するタテ [ヨコ] 列において、上 [左] 側外ワクに接するマスから始めて下 [右] 方向に1から順番に数えていった時、該当マスの相当する番号をいう。特に、タテヨコを区別しないとき、単に位置という。

(定義2) 外ワクか白マスで囲まれたスペースが、間に白マスがなく（これ以上細分化されない）、かつ1個以上の白以外のマスを含む時、区分といい、空きマスを含むと空き区分、すべてが黒マスなら閉じ区分、含まれるマスの数をその長さという。

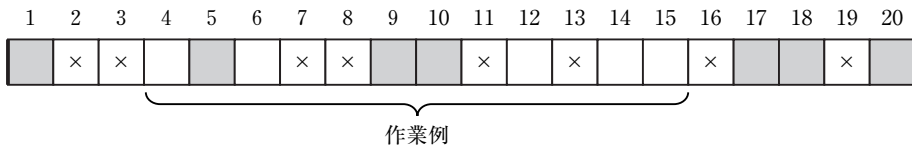
また、黒マスを少なくとも1個含む区分を黒有区分、すべてが空きマスのものを黒無区分という。

さて、各列に対し、「ワク決め」と呼ばれるつぎの操作を行う。

(定義3) 外ワクの上下ないし左右の両側から連続する黒白マス（解処理が済んだマス並び）を除いてできるスペースを作業列といい、この除外された部分を除外部という。さらに、間にある白ブロックを除くと、区分の並びができるが、これを区分並びという。

作業列及び各区分は、開始 [先頭] マス位置と長さで決まる。さらに、空き区分は、その各構成マスの黒・空きの状況で特徴づけられる。

例えば、下図はその一例である。



つぎに、各ピースの入れ場所に関する、その開始（先頭）マス位置と終了（尾）マス位置の限度値を導入する。

〔定義4〕 除外部に含まれるピースを除いたピース並びを、作業ピース並びという。

作業ピース並びに属する各ピースに対し、入れる場所に関して、その開始マス位置と終了マス位置の限度値－それ未満及びそれを超えることはありえない－を、それぞれ開始限度値、終了限度値という。

つまり、各ピースは開始限度値の位置をもつマスと終了限度値の位置をもつマスの間に入ることになる。

〔命題3〕（開始限度値と終了限度値）

作業列の始めのマス位置を  $START$ 、終わりのマス位置を  $LAST$  とする。作業ピース並びが長さそれぞれ  $p_1, p_2, \dots, p_k$  マスの  $k$  個のピースである時、各長さ  $p_i$  マスのピースの開始限度値  $PIT1(i)$  と終了限度値  $PIT2(i)$  は、つぎで与えられる。

① 作業列が一つの空き区分の場合：

$$PIT1(1) = START, \quad PIT1(i+1) = PIT1(i) + p_i + 1 \quad (1 \leq i \leq k-1)$$

$$PIT2(k) = LAST, \quad PIT2(i-1) = PIT2(i) - p_i - 1 \quad (k \geq i \geq 2)$$

② 作業列が  $n$  個の区分  $K_1, \dots, K_n$  をもつ場合：

$$PIT1(1) = START, \quad PIT2(k) = LAST。$$

$PIT1(i)$  が区分  $K_q$  に含まれるとして、 $PIT1(i+1)$  は、

・  $PIT1(i) + p_i + p_{i+1}$  の位置マスが  $K_q$  に含まれるなら、 $PIT1(i) + p_i + 1$ ,

・ 〳 〳 〳 ないなら、

区分  $K_{q+1}$  の先頭マスの位置 となる。

$PIT2(i)$  が区分  $K_r$  に含まれるとして、 $PIT2(i-1)$  は、

・  $PIT2(i) - p_i - p_{i-1}$  の位置マスが  $K_r$  に含まれるなら、 $PIT2(i) - p_i - 1$ ,

・ 〳 〳 〳 ないなら、

区分  $K_{r-1}$  の終わりマスの位置 となる。

この開始限度値と終了限度値を使うと、作業列が区分をもつ場合に「ピース入れ」処理が拡張できる。なお、証明は明らか。

〔命題4〕 作業ピース並びが長さそれぞれ  $p_1, p_2, \dots, p_k$  マスの  $k$  個のピースで、各長さ  $p_i$  マス

ピースの開始限度値と終了限度値を  $PIT1(i)$ ,  $PIT2(i)$  とする。

$PT1 = PIT1(i) + p_i - 1$ ,  $PT2 = PIT2(i) - p_i + 1$  と置けば、つぎが成り立つ。

もし  $PT2 \leq PT1$  なら、長さ  $p_i$  マスピースからその「ピース入れ」処理により、長さ  $(PT1 - PT2 + 1)$  の黒ブロックが生じ、その先頭マス位置は  $PT2$  である。

(例) 前にあげた長さ 13 マスの作業列が 2 つの長さ 6 マスの区分に分かれ、作業ピース並びが長さ 3 と 4 マスの 2 つなら、 $START=1$  として、 $PIT1(2)=8$ ,  $PIT2(2)=13$  から  $PT1=11$ ,  $PT2=10$  となり、長さ 2 の黒ブロック (先頭マス位置は 10) が長さ 4 マスのピースから生じる。

### 2.3 その他の重要な事項と処理

この節の命題 8 以下の主張で、マス位置が作業列を上〔左〕側外ワクからとる場合として述べられている。

#### 1) ピース可能数と可能タイプ

(定義 5) 黒ブロックについて、特定のピースの一部であるときピース確定ブロックといい、この特定ピースを確定ピースという。

ある区分内の黒ブロックを考える。この黒ブロックを一部として含む可能性をもつピースを可能ピースという。今、対象となるピースの集合に対し、可能ピースの集まりをとるとき、含まれるピースの個数を可能ピース数といい、可能ピース数が正のとき、含まれるピースの長さの最大値と最小値を可能最大値、可能最小値という。

また、可能タイプをつぎのように定める：

- ・可能ピース数が 0 の時、タイプ 0,
- ・可能最大値 = 可能最小値 (含まれるピースの長さがすべて等しい) なら、タイプ 1,
- ・それ以外をタイプ 2 とする。

[命題 5] 区分の長さを  $SL$ , 黒ブロックの長さとその先頭マス位置をそれぞれ  $BLG$ ,  $BITI$  とする。長さ  $PLG$  のピースは、つぎの性質をみたすとき可能ピースとなる。

- ①  $(PLG \geq BLG)$  かつ  $(PLG \leq SL)$ ,
- ② ピースの開始限度値と終了限度値を  $PIT1, PIT2$  とすれば、
  - (① を満たす) かつ  $(PIT1 \leq BITI)$  かつ  $(BITI + BLG - 1 \leq PIT2)$

(注意) 可能性をチェックする条件を強めれば強めるほど、可能ピースを絞り込むことができる。しかし、初級者のやり方は  $PIT1, PIT2$  を使わず①だけとみなせるので、解法のレベルに応じて①だけと①,②の共用を使い分ける。

(定義 6) ある区分において、そこで使用されるピースのスタート番号が一意に定まるとき使用ピース確定区分、そうでないとき使用ピース未確定区分という。

(定義 7) 各作業列の先頭区分と末尾区分を最端区分という。また、最端区分は使用ピース確定

であるので、その定まっているピースを最端ピースという。

明らかに、最端ピースは、作業ピース並びの上下または左右から見て先頭にくるものである。

[命題6] つぎのそれぞれの場合に、黒ブロックはピース確定となる。

① 最端区分での黒ブロックを考える。端から黒ブロックの先頭マスまでの空きマスの数を  $S1$ ，最端ピースの長さを  $PLG$  として、 $S1 < PLG + 1$  が満たされるとき。

② ある区分での黒ブロックで、その可能ピース数が1のとき。

[命題7] 可能タイプが1の時、黒ブロックの長さが可能最小値に等しいなら、ブロックのそれぞれの端に接触する空きマスは白マスとなる。

2) 「隣り合う2つの黒ブロックの結合と離反」

区分の壁に近い前者がピース確定ブロックであるとし、その確定ピース及び次に続くピースの長さを  $KPLG$ ,  $NPLG$  とする。また、2つの黒ブロックの長さを順に  $BLG1$ ,  $BLG2$ ，その間の空きマス数を  $S1$  とする。

[命題8]  $S1 = 1$  とする。間の空きマスは、その位置  $SITI$  が次の性質を満たせば、黒マスとなる：

( $BLG2 > NPLG$ ) 又は (位置( $SITI + NPLG + 1$ ))のマスが黒マス)。

(証明) 間の空きマスが白マスであると仮定すると、後者のブロックは確定ピースの次ピースの一部となる。ところがその先頭が壁と接触することから、長さ  $NPLG$  の黒マスブロックが生じ、また位置( $SITI + NPLG + 1$ )のマスは白マスで閉じられる。しかし、これは条件と矛盾する。

[命題9]  $BLG1 + BLG2 + S1 > KPLG$  なら、2つの黒ブロックは離反する(結合できない)。つまり、前者のブロックにおいて、壁に近い側で( $KPLG - BLG1 - S1 + 1$ )個の黒マスの伸びが生じる。

3) 「接触マスの白マス化」

使用ピース確定区分での壁に最も近い黒ブロックに対し、壁に近い側の端に接触するマスの白マス化を考える。

[命題10] 黒ブロックの長さを  $BLG$ ，壁からの空きマス数を  $S1$ ，壁に近い側の先頭黒マス位置を  $BITI$  とし、また、使用確定しているピースの番号を  $KP$  とする。

対象とするピースを  $KP$  から作業ピースの最後までとし、新しい黒ブロックとして先頭マス位置が( $BITI - 1$ )，長さが( $BLG + 1$ )のものをとり、その可能ピース数を  $KPN$  とする。次のいずれかの性質が満たされれば、壁に近い側の接触マスは白マスとなる。

① 可能ピース数が0である，

② 可能ピースの最小番号を  $P1$  とするとき、 $KP < P1$  であり、 $P1 - KP = n$  とする。

$KP$  から ( $P1 - 1$ ) までのピース長さの総和を  $TPLG$  として、 $TPLG + n - 1 > S1 - 2$ 。

(証明) 黒ブロックの壁に近い側の接触マスを黒マスと仮定する。すると、この黒ブロックとの



仕切りを考慮した長さ  $(S1 - 2)$  のスペースに、KP から少なくとも  $(P1 - 1)$  までのピースが入らなければならないが、条件からこれは不可能となるから。

[系1] 最端区分での黒ブロックを考える。最端ピースを含め長さ  $PLG$  のピース並びが  $n$  個続いているとする。BLG, S1 を黒ブロックの長さとして端からの空きマス数とするとき、つぎの性質が満たされれば命題10の主張が成り立つ。

$$(BLG=PLG) \text{ かつ } (n*PLG + n - 1 > S1 - 2).$$

(注意) 命題2の前半の主張は、この系1の特別のケースである ( $PLG=1$ )。

#### 4) 「1ピース入れ」と「2ピース入れ」

(定義8) 使用ピース確定区分での黒ブロックを考える。使用されるピース並びの1番目, 2番目をとる時、1番目だけしかこの区分に入れられないとき1ピース区分、入るピースが1番目と2番目だけの丁度2個であるとき2ピース区分という。

[命題11] 区分の長さを  $SL$ , 使用されるピース並びの1番目, 2番目, 3番目のそれぞれの長さを  $PLG1, PLG2, PLG3$  とする。

①  $SL < PLG1 + PLG2 + 1$  なら、1ピース区分である。

② 区分に含まれる黒マス群の、一方の端の位置を  $ITI1$ , 他方の端の位置を  $ITI2$  とする (黒ブロックが複数の時、1番目の先頭マス位置と最終ブロックの末尾マス位置)。

$$(ITI2 - ITI1 + 1 > PLG1) \text{ かつ } (SL < PLG1 + PLG2 + PLG3 + 2)$$

なら、2ピース区分である。

## 2.4 解手法のリスト

解法のための手法は大きく5つのタイプに分けられる。

### 1) LOCAL 処理

各作業列の両方向からみた最端区分における処理である。

最端区分の長さを  $SL$ , 最端ピースの番号と長さを  $HP, PLG$  とする。また、黒ブロックのピース可能数を求める時、特に断らない限り、対象とするピースは作業ピース全体とする。なお、黒ブロックの先頭マス位置は端から測るものとする。

(A) 黒有区分に関するもの

[手法1] 最端ピースの次ピースの長さを  $NPLG$  とする。

①  $2PLG > SL$  なら、最端ピースによる「ピース入れ」処理を行なう。

②  $SL < PLG + NPLG + 1$  なら、最端ピースによる「1ピース入れ」処理を行なう。

[手法2] 端に最も近い黒ブロックを考え、その長さを  $BLG$ , 端からの空きマス数を  $S1$ , 壁に近い側の先頭黒マス位置を  $BITI$  とする。

① その可能タイプが1の場合

- ・  $BLG =$  可能最小値なら、命題7を適用する。
- ・ 特に、可能数が1（確定ピース番号を  $KP$  とする）の時、つぎの処理を行なう。
  - (i)  $HP = KP$  なら、「黒マス伸び・白マス埋め」処理。
  - (ii)  $HP \neq KP$  なら、端からの長さ  $(S1 - 1)$  のスペースをとり、 $HP$  から  $(KP - 1)$  番までのピースによる「ピース入れ」処理。

②  $PLG = BLG$  の場合

新しい黒ブロックとして先頭マス位置が  $(BITI - 1)$ 、長さが  $(BLG + 1)$  のものを取り、命題10を用いた「接触マスの白マス化」処理を行なう。

[手法3] 黒ブロックが複数ある場合で、端に最も近いものと次のブロックだけを考える。前者がピース確定ブロックであるとし、2つのブロックの長さを順に  $BLG1, BLG2$ 、間の空きマス数を  $S2$ 、そして確定ピースの長さを  $KPLG$  とする。

- ①  $S2 = 1$  なら、「隣り合う黒ブロックの結合」処理を行なう。
- ②  $BLG1 + S2 + BLG2 > KPLG$  の場合
  - ・ 「隣り合う黒ブロックの離反」を行なう。
  - ・ 後者ブロックの可能タイプが1のとき、つぎの処理を行なう。
    - (i)  $BLG2 =$  可能最小値なら、命題7を適用する。
    - (ii) 可能数が1（確定ピースの番号と長さを  $NKP, NPLG$  とする）の時、
      - a)  $KP + 1 = NKP$  なら、後者ブロックに関する「隣り合う黒ブロックの離反」及び両者の伸びでも届かない中間部を白マスにする。
      - b)  $NKP - KP \geq 2$  なら、長さ  $(S2 - 2)$  の中間スペースに、 $(KP + 1)$  から  $(NKP - 1)$  までのピースによる「ピース入れ」処理を行なう。

[手法4] 問題としている端の反対側の壁から、その壁に最も近い黒ブロックに対して、ピース可能情報をもとに「黒マス伸び、白マス埋め」処理を行なう。

すなわち、このブロックの長さとして可能最小値を  $RBLG, MNKLG$ 、反対壁との間の空きマス数を  $S9$  とするとき、

- ①  $MNKLG > S9 + RBLG$  なら、ピースの長さを  $MNKLG$  とし「黒マス伸び」処理を行なう。
- ② 可能タイプが1の場合
  - ・  $MNKLG = RBLG$  なら、命題7を適用する。
  - ・ どの可能ピースに対しても、その次のピースが  $S9$  内に入れられないなら、ピースの長さを  $MNKLG$  として「白マス埋め」処理を行う。

[手法5] 対象とする黒ブロックの性質は、次に続く区分における黒マス情報によっても決まる。そこで、次の区分における壁に最も近い黒ブロックを考え（間の空きマス数を  $NS$  とする）、そ

のピース可能数が1（確定ピース番号を KP とする）のとき、

- ①  $HP+1=KP$  なら、最端区分で最端ピースによる「1ピース入れ」処理を行なう。
- ② 長さ  $(NS - 1)$  のスペースに、 $(KP - 1)$  番から HP 番までのピースを逆順に入れ、最端区分において残ったピースによる「ピース入れ」処理を行なう。

(B) 黒無区分に関するもの

[手法6] 手法5をこの場合にも使う。まず、最端区分に最も近い黒有区分を見つける。この区分における壁に最も近い黒ブロックを考え、そのピース可能数が1（確定ピース番号を KP とする）のとき、

- ①  $HP=KP$  なら、最端区分を含む直前までの黒無区分をすべて白マスにする。
- ② この区分が最端区分の次である場合、手法5②を行なう。

2) 準-LOCAL 処理

LOCAL 処理は最も取扱いやすい最端区分を対象にする。しかし、例えば、最端区分が「1ピース入れ」か「2ピース入れ」なら、それが完全に確定していなくても、それを飛ばして、次の区分に LOCAL 処理が適用できる (LOCAL 処理の大事なポイントは使用ピース確定性である)。

(A) 黒有区分に関するもの

[手法7] 最端区分が「1ピース入れ」か「2ピース入れ」の場合、この最端区分を飛ばして縮められた作業列で、LOCAL 処理を行なう。

(B) 黒無区分に関するもの

長さ1マスのピース並び（個数を  $PLG1SU$  とする）及びその次のピース（番号を  $NPBAN$ 、長さを  $NXPLG$  とする）を考える。最端区分内にこれらのピース全部が入れられないとき、続く区分にこのピース情報が伝わる。それと区分続きの性質を利用した処理である。

[手法8] 最端区分内に長さ1マスのピース並び全部が入れられないとする。

最初の黒有区分をとり、この区分における壁に最も近い黒ブロック（長さを  $B1$ 、壁との間の空きマス数を  $S1$  とする）を考える。

さて、直前までのすべての黒無区分（最端区分を含む）に長さ1マスのピースを入れるだけいれ、まだ  $NOKORI1$  個のピース並びが残っているとき、

- ① この区分で命題2を適用する、
- ②  $B1 \geq 2$  かつ長さ  $(S1 - 1)$  の空きスペースに丁度  $NOKORI1$  個だけの1マスピースが入るなら、それまでの黒無区分すべてと長さ  $(S1 - 1)$  の空きスペースとに長さ1のピース並びを入れられるだけいれる。

[手法9] 最端区分内に  $NPBAN$  番のピースが入れられないとする。

このピースが入れられる最初の区分（つまり、区分の長さが  $NXPLG$  以上）までの間に、見

えている長さ1の黒ブロックの合計数がPLG1SUと一致したなら、これらの黒ブロックは長さ1ピースと確定し、かつ途中の黒無区分（最端区分を含む）はすべて白マスになる。

### 3) GLOBAL 処理

(A) すべての区分にわたる黒ブロックに関するもの

[手法10] 各区分の黒ブロックに対して、手法2～手法4を適用する。但し、使用ピース確定性が使えない点に注意する。

(B) ある性質をもつ区分の数に関するもの

[手法11] 作業列における黒有区分の数をKBLSU、作業ピースの数をBMSUNとする。

① KBLSU= BMSUNならば、黒有区分は1ピース区分、黒無区分はすべてのマスが白マスとなる。

② KBLSU= BMSUN - 1の場合

(i) KBLSU=1で、この黒有区分が次の性質(P)をもつ複数の黒ブロックを含むとき、黒有区分は2ピース区分、黒無区分はすべてのマスが白マスとなる。

(ii) 黒有区分の中に次の性質(P)をもつ複数の黒ブロックを含むものがあるとき、この黒有区分は2ピース区分、他の黒有区分は1ピース区分、かつ黒無区分はすべてのマスが白マスとなる。

(P) 「これらの黒ブロックを一緒にまとめてできる黒ブロックの長さが、それを一部として含む可能性のある2つの作業ピースの長さより大きい」

[手法12] 作業ピースの長さについて、1つだけ2以上のもの（長さをPLGとする）があり、他はすべて長さ1とする。ある黒有区分が2以上の長さの黒ブロックを含むなら、次が成り立つ：

① それ以外の区分に含まれる長さ1の黒マスは長さ1のピースとして確定し、

② 同じ区分に含まれる長さ1の黒マスは、問題の黒ブロックに接続されない（つまり、一緒にまとめてできる黒ブロックの長さがPLGを超える）なら、長さ1のピースとして確定する。

[手法13] 最端区分からはじめて、最初の未確定の黒有区分（長さをSLとする）をとる。この区分が使用ピース可能区分（そのピース番号をKB、次に続くピースをNKB番とし、それぞれのピースの長さをPLG1、PLG2とする）で、この区分に含まれる唯一の黒ブロックが反対側の壁に接触しているとき、次が成り立つ：

黒ブロックの可能ピースがKBとNKB番の2つで、 $PLG1 < PLG2$ なら、壁から  $(SL - PLG2)$  の位置のマスは白マスとなる。

(注意) ① もし黒マスなら、この黒マスはKBとNKBどちらのピースの一部にもなりえない

から。

② 手法 13 の条件を満たす黒ブロックを探す途中で、閉じた黒ブロックがある場合、その可能ピース数が 1 なら、この操作を継続できる。

(C) 開始限度値と終了限度値の縮め修正

[手法 14] 作業列内の黒ブロックの一つ一つについて、その可能ピースに応じた形で該当するピースの開始限度値と終了限度値を直す。さらに、それに伴って関係するピースの開始限度値と終了限度値を修正する。

4) タテ列と横列の複合処理

上の 3 つの処理は、タテ列と横列のどちらか一方だけの黒白マス情報を用いた処理であった。ここでは、発生した黒マス並びに伴って直角方向に伸び出した黒マス群に関する元の列方向からの処理、つまりタテ列と横列の複合処理を扱う。すなわち、このようにして伸び出した黒マス群に対して、スタートとなった列（発生した黒マス並びを含む）から始めてある列数を取り、各列におけるタテ・ヨコ方向から見たとき、この列の作業ピース並びとの間に矛盾が生じないか調べ、それに応じた措置をとる。


(A) 目処理

ある状況を想定したとき、“目だけで追える処理”を「目処理」と呼ぶ。

ここでは、解処理が完全に済んだ列を除いた外ワクに最も近い 4 つの辺をとる。この 4 辺の両側からの最端区分の先頭マスを選び、それが黒マスと仮定した時に発生する黒マス並びに応じて直角方向に伸び出した黒マス群を扱う。また、辺から近い 2 列だけについて（実際の作業で“目だけで追える”対象と考える）調べる。

[手法 15] この場合に、その作業ピース並びとの間に矛盾が起きたなら、選ばれた先頭マスは白マスになる。

(B) 特定区分での複合処理

ケース 1：右図のように、両側に同数でかつ複数の空 ×  ×  
きマス並びをもつ黒ブロックの 1 ピース区分をと

る。一方の接触マスを白マスとすれば反対側に複数の黒マス並びが発生する。

ケース 2：使用ピース確定であり、使用確定であるピースの挿入により複数の黒マス並びが発生する、そんな黒無区分をとる。

(i) 挿入したときの発生した黒マス並びを対象にする。

(ii) ピースが入らないとすると空きマスすべてが白マスになるが、その結果の白ブロックの両面に接触する黒マス並びを対象にする。

なお、選ばれる列数は直角方向に伸び出した黒マスブロックの長さの最大値とする。

[手法 16] 矛盾が生じたとき、

① ケース 1 では、白マスと仮定した接触マスは黒マスとなる。

② ケース 2 では、(i) の場合は黒無区分をすべて白マスとし、(ii) の場合は「ピース入れ」処理を行なう。

(C) 両方の接触マスから伸び出す黒マスへの手法 11 の適用処理

両側に同数でかつ 1 個以上の空きマス並びをもつ黒ブロックの 1 ピース区分をとる。黒マス並びの両方の接触マス、その少なくとも一方は黒マスとなるので、この両方を黒マスとしたとき生じる直角方向に伸び出す 2 つの黒マス並びに対し、2 つの伸びが共通する部分を対象列として考える。

つまり、この列を元のタテ・ヨコ方向から見れば、少なくとも 1 個の黒マスが増えることになる。この増えた黒マスに応じた形で手法 11 を適用するのである。

[手法 17] 黒マス増加の 2 つのいずれの場合にも適用した手法 11 の条件が満たされるなら、増えた黒マス (2 つの可能性あり) を含む黒有区分以外は、手法 11 と同じ処理を行なう。さて、増えた黒マスを含む黒有区分では、

手法 11 の条件①が成り立つ場合：制限付き 1 ピース入れ－黒マスを増やす措置だけをやめる－を行なう。

手法 11 の条件②が成り立つ場合：その区分が、

(i) 1 個の黒マスを含むまたは 2 個の黒ブロックを含むが性質 (P) を満たさないときは、制限付き 1 ピース入れを行なう。

(ii) 2 個の黒ブロックを含み、かつ性質 (P) を満たすとき、制限付き 2 ピース入れ－黒マスを増やす措置だけをやめる－を行なう。

#### 5) 矛盾処理

ある空きマスを黒か白と仮定し、このもとで処理を続けた結果矛盾が起ったら、そのマスを逆の色として決める処理であるが、選ばれる対象となる空きマスは次のように定める：

解処理が完全に済んだ列を除いた外ワクに最も近い 4 つの辺から始めて、段々離れた列をとっていく。各列では、① 両側からの最端区分の先頭マス、② 最端区分における先頭から最端ピースの長さ分離れた位置での白マス、③ すべての区分の先頭マスと末尾マス、という順にレベルに応じて選んでいく。

[手法 18] 選択マス候補をリストとして作成する。リスト順に選択マスを選び、それを指定した色 (黒か白) と仮定し、その追加情報をもとに処理を進め、矛盾が起ったら、選んだマスを指定した色と逆の色にする。

### 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

参考に用いた [2], [3], [4] に基づき、レベル1～レベル5までの5段階とする。

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、「イラストパズル」においても、問題が本来持っている難しさ（論理思考の高い低い）に加えて、解くときの手間とか面倒くささを考慮する。ところで、論理的な解き方は2.4で手法として示したが、これら手法のレベルの高低は、

- ① その原理（アイデアと論理性）の発見のむずかしさ、と
- ② その手法を実行するときの困難さ

の2つの要素で決まると考えられる。例えば、解手法を5つのタイプにわけたが、局所性から大域性への広がり②と深いつながりを持ち、タテ・ヨコの複合処理は①との関連が強く、さらに「目処理」と一般的な矛盾処理との違いは②と深い関係をもつと考えられる。

さて、難易度の基準化をつぎの2点にまとめる。

1) 論理思考の高低にもとづく具体的手法によって決まるレベルに、解くときの手間等をプラスさせる。

2) 手法のレベルの高低は、つぎのようにする。

- ① 2.1で示した4つの基本処理は、解くための核でありレベル1とする、
- ② 調べる作業回数が多いものほど、レベルを高くし、
- ③ 拠点をもとにした局所的な処理は低い、拠点から離れたものやグローバルな視点から考えるものは高い、
- ④ 気づきにくい特殊状況（発見のむずかしさ）はレベルが高く、
- ⑤ 矛盾処理－用意した典型的な手法がだめなときの切り札となる－は、確実性も低く手間がかかるのでレベルを高くするのが妥当であるが、実際に問題を解く際、未確定のマス数が少なければ強引に行なわれる。これを踏まえて、レベルと未確定のマス数に応じた考慮を行なう。
- ⑥ 同じ手法でも、処理内容によって、つまり先にあげた2つの要素を考慮して、レベルに違いをもたせる。また、適用する範囲もレベルによって違いをもたせる。

例えば、作業列全体での「ピース入れ」処理に関して、その対象となる列は拠点の列（外ワクを基点として解処理が済んだものを除いていく）からある幅で選ぶが、その幅はレベルが高くなるほど大きくする。

### 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
FIRSTCLUE	最初の糸口（ピース入れ）	レベル 1
KIHONSYORI(a)	4つの基本処理	〃
RETUKANSEI	手法 11 の適用	〃
WAKUSYUKUSYO	区分全体にわたる白マス埋め	〃
SPEPIECEIRE	「ピース入れ」処理	〃
KIHONSYORI(b)	手法 1	レベル 2
HANAREMASUSYORI1	手法 2	〃
REVSIDEMSNOB I	手法 4	〃
FIRSTBMASU	手法 6	〃
KYOTUKUROMASU	命題 4 の適用	〃
HUKUSUMASUSYORI	手法 3	レベル 3
HANAREMASUSYORI2	手法 2 の適用	〃
NEXTKUBUNMASU	手法 5	〃
PRELOCALSYORI	手法 7	〃
PLG1SYORI2	手法 8	〃
KMASULENSYORI	手法 10	〃
MESYORI	手法 15	〃
PLG1SYORI3	手法 9	レベル 4
TOTALMXLG2SYORI	手法 12	〃
NOKUROITI	手法 13	〃
RETUKANSEI2	手法 11	〃
PITITIJIME	手法 14	〃
TAYOHUKUGOSYORI	手法 15～手法 17	レベル 5
MUJYUNSYORI	矛盾処理	レベル 3～5

なお、多くの手法の中に、様々なレベルのケースが含まれている。

### 3.3 解くときの手間や面倒くさを反映させる基準

‘調べる作業回数の多い少ない’が基本であるが、対象となる作業は各レベルにマッチしたものでなければならない。そこで、解法の手間を反映する尺度として、解レベルを決定付ける手法処理を何回行なったかをとる。つまり、解レベルが3となったとき、黒白マスの決定にお



いて、レベル3の手法を何回使ったかをカウントし、その数によって基準を定める。

これについては、5章において具体的問題を解く中でなるべく合理的な形で決める。

#### 4. レベルを計算するプログラム

```
rem *** ILLUSTRATION ***
def fnDR(CS)=((TY+2*CS-1) mod 4)+1
'
dim EPL(1,210,50),EMA(210,210),PITI(50,2),KASE(1,210),CH(1,210),GR(1)
dim PRL(100),ETI(210),SP(210),WKL(100),GEN(100),BMA(100),TML(100),KIT(100,2)
dim HLT(200),KANOLT(100),BMAL(100),BNLT(50),BNIT(50),SNLT(50),SNIT(50)
dim MWKL(100),MGEN(210),MBMA(100),MTML(100),MPITI(50,2)
dim RIT(100),KPLT(100),MNSYO(100),STKR(100),BSUL(100,2)
dim KBL(2,50),BRK(2,50),SP1(200),SP2(200),MX(50),MN(50),SKRL(30)
dim KHL(3000,7),HEMA(210,210),HKASE(1,210)
dim MAXDAN(5),MAXPSU(5),KYCUTSU(5),MXHANARE(5),ZT(4),ZY(4),RD(4)
dim MNHT(1,5),UPKIYUN(5),LVKSU(5)
' -- data_yomikomi --
gosub *DATAYOMI
' -- syoki_settei --
for I=1 to GR(0): EMA(0,I)=-1: EMA(GR(1)+1,I)=-1: next I
for J=1 to GR(1): EMA(J,0)=-1: EMA(J,GR(0)+1)=-1: next J
MAXDAN(1)=3:MAXDAN(2)=6:MAXDAN(3)=15:MAXDAN(4)=30:MAXDAN(5)=150
MAXPSU(1)=3:MAXPSU(2)=6:MAXPSU(3)=10:MAXPSU(4)=20:MAXPSU(5)=50
KYCUTSU(1)=5:KYCUTSU(2)=4:KYCUTSU(3)=3:KYCUTSU(4)=2:KYCUTSU(5)=1
MXHANARE(2)=2:MXHANARE(3)=4:MXHANARE(4)=6:MXHANARE(5)=50
for I=1 to 4: read ZT(I),ZY(I): next I
data -1,0, 0,1, 1,0, 0,-1
RD(1)=3:RD(2)=4:RD(3)=1:RD(4)=2
MNHT(0,3)=5:MNHT(1,3)=3: MNHT(0,4)=20:MNHT(1,4)=10: MNHT(0,5)=100
UPKIYUN(1)=20000:UPKIYUN(2)=100:UPKIYUN(3)=50:UPKIYUN(4)=25:UPKIYUN(5)=20000
' --- KAI_syori ---
LEVEL=1
FEND=0: FKAN=0
while FEND=0
  for I=1 to 5: LVKSU(I)=0: next I
  FEND=1
  ' -- hairetu_syoki-ka --
  for I=1 TO GR(0):for J=1 TO GR(1): EMA(J,I)=0: next J,I
  for TY=0 TO 1:for G=1 TO GR(TY): KASE(TY,G)=0:CH(TY,G)=0: next G,TY
  ' -- first_clue --
  LVK=0: gosub *FIRSTCLUE
  ' -- tujoyo_syori --
  FMUSY=0: FMUREN=0
  *TUJYOSYORI
  *MUSYOMODOSI
```

```

FCHECK=1: FMU=0
while FCHECK=1
FCHECK=0
for TY=0 to 1
' --- UDLRDAN ---
gosub *UDLRDAN
if FKANSCH=1 then AKIMN=0:goto *KANSCH
' --- gyo(retu)_syori ---
for G=ULDAN TO DRDAN
if (CH(TY,G)=1)AND(KASE(TY,G)=0) then
CH(TY,G)=0: FGYOSYO=0: FPITIHEN=0
WAKUTY=0: FSIBORI=0: FMESY=0 ' main_param-ti
' -- waku & pieceiti_kime --
gosub *ETISAKUSEI
gosub *WAKUKIME
if FMUSY=1 then gosub *MUARCHI 'mujun_ch
if FMU=1 then *MUJYUN
RLEN=GR(1): if GR(0)<RLEN then RLEN=GR(0)
MXKYORI=int(RLEN/KYCUTSU(LEVEL))
gosub *PIECEITI
*PRELOCAL
gosub *MAXMINLG ' TOTALMXNLG
*NEWPITI
' -- basic_syori --
HSU=0: CS=1
if FGYOSYO=0 then gosub *RETUKANSEI
if FGYOSYO=0 then gosub *WAKUSYUKUSYO
if FGYOSYO=0 then gosub *SPEPIECEIRE
if FGYOSYO=0 then gosub *KYOTUKUROMASU: FPITIHEN=0
' -- local_syori --
if FGYOSYO=0 then gosub *LOCALSYORI
' -- pre-local_syori --
FPRELC=0
if (FGYOSYO=0)and(FMUSY=0)and(LEVEL>=3)and(NPN>=2)and(BMASUN>=2) then gosub *PRELO-
CALSYORI
if FPRELC=1 then *PRELOCAL
' -- global_syori --
if (FGYOSYO=0)and(LEVEL>=3) then
gosub *WAKUKIME: gosub *MAXMINLG
HSU=0: gosub *GLOALSYORI
if FPITIHEN=1 then *NEWPITI
end if
end if
next G,TY
wend
' -- kansei_hantei --
gosub *AKIMASUSU
*KANSCH

```

```

if (FMUSY=0)and(AKIMN=0) then FKAN=1:goto *KANS
' -- tate-yoko_hukugo-syori --
FTYHUKUGO=0
if (LEVEL>=3)and(FMUSY=0)then gosub *MESYORI
if (LEVEL>=4)and(FMUSY=0)and(FTYHUKUGO=0) then gosub *TAYOHUKUGOSYO
if FTYHUKUGO=1 then *TUJYOSYORI
' -- mujyun_ari-check --
if (FMUSY=1)and(FMU=0) then
  gosub *MUARICH
  if (AKIMN=0)and(FMU=0) then FKAN=1:goto *KANS
end if
' -- mujyun_syori --
*MUJYUN
if LEVEL>=3 then
  TMASU=GR(0)*GR(1):NRHIRITU=int(AKIMN/TMASU*100+0.5)
  if (NRHIRITU<=MNHT(0,LEVEL))or((TMASU>=2500)and(NRHIRITU<=MNHT(1,LEVEL))) then
    gosub *MUJYUNSYORI
    if FMSTOP=0 then *MUSYOMODOSI
  end if
end if
' -- LEVEL_up --
if LEVEL<5 then LEVEL=LEVEL+1: FEND=0
wend
*KANS
' -- LEVEL_up --
if LVKSU(LEVEL)>=UPKIJYUN(LEVEL) then LEVEL=LEVEL+1
if (LEVEL=1)and(GR(0)*GR(1)>=2000) then LEVEL=LEVEL+1
' -- LEVEL_hyoji --
locate 40,20: print "LEVEL=";LEVEL
end
:::
*DATAYOMI
OPEN "a:PROBLEM.dat" for INPUT AS #1
PPROB=1:RFILE=0
while NOT EOF(1)
  INPUT #1, GR(1),GR(0)
  for K=0 TO 1:for I=1 TO GR(K)
    INPUT #1,DN: EPL(K,I,0)=DN
    for J=1 TO DN: INPUT #1,EPL(K,I,J): next J
  next I,K
wend
CLOSE #1
return
::
*FIRSTCLUE
for TY=0 TO 1
  SITI=1:SL=GR(1-TY)
  for G=1 to GR(TY)

```

```

if EPL(TY,G,0)=0 then
  for R=1 to SL
    if TY=0 then EMA(R,G)=-1 else EMA(G,R)=-1
    CH(1-TY,R)=1
  next R
  KASE(TY,G)=1
else
  STBAN=1:LTBAN=EPL(TY,G,0):gosub *MAXMINLG
  ULDAN=1:DRDAN=GR(TY):CS=1:gosub *ETISAKUSEI: gosub *RETUPIECEIRE
end if
next G,TY
return
::
*RETUPIECEIRE
' -- par. ULDAN,DRDAN, TOTALMXNLG, STBAN,LTBAN, CS,SL,SITI --
PN=LTBAN-STBAN+1
' -- tekiyo_hantei --
FGO=0: MXDN=MAXDAN(LEVEL)
if PN<=MAXPSU(LEVEL) THEN
  if (G<ULDAN+MXDN)OR(G>DRDAN-MXDN) THEN
    FGO=1      ' kanarazu_suru
  else        ' jyoken-tuki_suru
    if (LEVEL=1)and(SL<=(2*TOTALMXLG+TOTALMXLG2)) then FGO=1 ' case_A
    if (LEVEL>=2)and(SL<=(2*TOTALMXLG+TOTALMXLG2+PN)) then FGO=1 ' case_B
    PWA=0      ' piece-nagasa_sowa
    for IJ=STBAN to LTBAN: PWA=PWA+EPL(TY,G,IJ): next IJ
    if (LEVEL>=3)and((PWA+PN)>=SL-2) then FGO=1 ' case_C
  end if
end if
' -- piece_ire --
if FGO=1 THEN
  gosub *SPSAKUSEI ' SP_sakusei
  for IK=1 TO PN: PRL(IK)=EPL(TY,G,STBAN+IK-1): next IK
  HSU=0: gosub *PIECEIRE
  if KIT(PN,1)=SL THEN KASE(TY,G)=1
  if HSU>=1 then gosub *EMAKAKIKAE ' EMA_kakikae
end if
return
::
*PIECEIRE
' -- par. SP(), PN,PRL( ) --
' KIT_sakusei
PIS=-1: PIL=SL+2
for IK=1 TO PN
  PIS=PIS+1+PRL(IK): PIL=PIL-1-PRL(PN+1-IK)
  KIT(IK,1)=PIS: KIT(PN+1-IK,2)=PIL
next IK
' kuro-masu_ume

```

```

for IJ=1 TO PN
  MSTI=1:UMEIT1=KIT(IJ,2):UMEIT2=KIT(IJ,1): gosub *KUROSIROMASUUME
next IJ
' sikiri_siro-masu
if KIT(PN,1)=SL THEN
  GITI=0
  for IJ=1 TO PN-1
    GITI=KIT(IJ,1)+1
    if SP(GITI)=0 THEN SP(GITI)=-1:HSU=HSU+1:HLT(HSU)=GITI
  next IJ
end if
return
::
*UDLRDAN
FKANSCH=0: ULDAN=-1
for I9=1 to GR(TY)
  if KASE(TY,I9)=0 then ULDAN=I9: exit for
next I9
if ULDAN=-1 then FKANSCH=1: return
for I9=GR(TY) to 1 step -1
  if KASE(TY,I9)=0 then DRDAN=I9: exit for
next I9
return
::
*ETISAKUSEI
' -- par. TY,G --
for IK=0 to GR(1-TY)+1
  if TY=0 then ETI(IK)=EMA(IK,G) else ETI(IK)=EMA(G,IK)
next IK
return
::
*WAKUKIME
' -- par. TY,G, WAKUTY --
FCOUT=0:FTUME=1:FBMARI=0: WKN=0
for I9=1 TO GR(1-TY)+1
  if ETI(I9)<>-1 THEN
    if FCOUT=0 THEN FCOUT=1: STIT=I9
    if ETI(I9)=0 THEN FTUME=0 else FBMARI=1
  else
    if FCOUT=1 THEN
      WKN=WKN+1: WKL(WKN)=I9-STIT
      GEN(WKN)=STIT: TML(WKN)=FTUME: BMA(WKN)=FBMARI
      FCOUT=0:FTUME=1:FBMARI=0
    end if
  end if
next I9
if WAKUTY=1 then return
' START,LAST_jyoho

```

```

FRKAN=1
for I9=1 TO WKN
  if TML(I9)=0 THEN FRKAN=0: EXIT for
next I9
if FRKAN=1 then KASE(TY,G)=1:STBAN=0:LTBAN=-1: FGYOSYO=1:return
STPAN=I9:STBAN=I9:START=GEN(I9)
'

for I9=WKN TO 1 STEP -1
  if TML(I9)=0 THEN EXIT for
next I9
LTPAN=I9:LTBAN=EPL(TY,G,0)-(WKN-I9):LAST=GEN(I9)+WKL(I9)-1
NPN=LTPAN-STPAN+1: BMASUN=LTBAN-STBAN+1
return
::
*PIECEITI
if LEVEL<=2 THEN
  PIS=START:PIL=LAST
  for IK=STBAN TO LTBAN
    PITI(IK,1)=PIS: PITI(LTBAN+STBAN-IK,2)=PIL
    PIS=PIS+EPL(TY,G,IK)+1: PIL=PIL-EPL(TY,G,LTBAN+STBAN-IK)-1
  next IK
else ' LEVEL>=3
  ZKAKUITI=START:ZKAKUSB=STPAN:MBAN=STBAN: gosub *PITI1
  ZKAKUITI=LAST :ZKAKUSB=LTPAN:MBAN=LTBAN: gosub *PITI2
end if
return
::
*PITI1
' -- par. ZKAKUITI,ZKAKUSB,MBAN --
for IJ=MBAN to LTBAN
  for IK=ZKAKUSB to LTPAN
    BLITI=ZKAKUITI+EPL(TY,G,IJ)-1: SLITI=GEN(IK)+WKL(IK)-1
    if BLITI<=SLITI THEN EXIT for else ZKAKUITI=GEN(IK+1)
  next IK
  PITI(IJ,1)=ZKAKUITI
  ZKAKUITI=BLITI+2: ZKAKUSB=IK
next IJ
return
::
*PITI2
' -- par. ZKAKUITI,ZKAKUSB,MBAN --
for IJ=MBAN TO STBAN STEP -1
  for IK=ZKAKUSB TO STPAN STEP -1
    BLITI=ZKAKUITI-EPL(TY,G,IJ)+1: SLITI=GEN(IK)
    if BLITI>=SLITI THEN EXIT for else ZKAKUITI=GEN(IK-1)+WKL(IK-1)-1
  next IK
  PITI(IJ,2)=ZKAKUITI
  ZKAKUITI=BLITI-2: ZKAKUSB=IK

```

```

    next IJ
    return
    ::
*MAXMINLG
TOTALMXLG=0:TOTALMXLG2=0: TOTALMINLG=1000
for IK=STBAN TO LTBAN
PPL=EPL(TY,G,IK)
' MAX
if PPL>TOTALMXLG THEN
TOTALMXLG2=TOTALMXLG: TOTALMXLG=PPL
else
if PPL>TOTALMXLG2 THEN TOTALMXLG2=PPL
end if
' MIN
if PPL<TOTALMINLG THEN TOTALMINLG=PPL
next IK
return
::
*KUROSIROMASUUME
' -- par. MSTI, UMEIT1, UMEIT2 --
if (UMEIT1<0)or(UMEIT2>200) then return      ' mujyun_check
'
for IK=UMEIT1 to UMEIT2
if SP(IK)=0 THEN SP(IK)=MSTI:HSU=HSU+1:HLT(HSU)=IK
next IK
return
::
*RETUKANSEI
LVK=1
gosub *KBLOCKSU
if (KBLSU=BMASUN)and(KBLSU-BKANSU<MAXPSU(LEVEL)) then BRKBN=1000:gosub *KUBUNSY-
ORI
return
::
*KBLOCKSU
KBLSU=0: BKANSU=0
for I9=STPAN to LTPAN
if BMA(I9)=1 then KBLSU=KBLSU+1:BMAL(KBLSU)=I9
if TML(I9)=1 then BKANSU=BKANSU+1
next I9
return
::
*KUBUNSYORI
' -- par. BRKBN, KBLSU --
' siro-masu_ume
for I9=STPAN TO LTPAN
if BMA(I9)=0 THEN HSU=0: SBAN=I9:gosub *KUBUNSIROUME
next I9

```

```

' piece-ire_syori
for I9=1 to KBLSU
  HSU=0: SL=WKL(BMAL(I9)):SITI=GEN(BMAL(I9)): gosub *SPSAKUSEI
  if I9=BRKBN then SBAN=BMAL(I9):PLG1=GPICN1:PLG2=GPICN2: gosub *PIECE2SET
  if (I9<>BRKBN)and(TML(BMAL(I9))=0) then
    if I9<BRKBN then PLG=EPL(TY,G,STBAN+I9-1) else PLG=EPL(TY,G,STBAN+I9)
    gosub *PIECE1SET
  end if
  if HSU>=1 then gosub *EMAKAKIKAE: FGYOSYO=1 ' EMA_kakikae
next I9
return
::
*KUBUNSIROUME
' -- par. SBAN,CS --
if BMA(SBAN)=1 then FMU=1: return ' jyogai
SL=WKL(SBAN): if CS=1 then SITI=GEN(SBAN) else SITI=GEN(SBAN)+SL-1
gosub *SPSAKUSEI
MSTI=-1:UMEIT1=1:UMEIT2=SL:gosub *KUROSIROMASUUME
gosub *EMAKAKIKAE: FGYOSYO=1
return
::
*PIECE1SET
' -- par. SL,PLG,SP() --
gosub *PB12ITI
' kuromasu_oki
MSTI=1:UMEIT1=PB1:UMEIT2=PB2: gosub *KUROSIROMASUUME
  UMEIT1=PB2:UMEIT2=PLG: gosub *KUROSIROMASUUME
  UMEIT1=SL-PLG+1:UMEIT2=PB1: gosub *KUROSIROMASUUME
' siromasu_ume
MSTI=-1:UMEIT1=1:UMEIT2=PB2-PLG: gosub *KUROSIROMASUUME
  UMEIT1=PB1+PLG:UMEIT2=SL: gosub *KUROSIROMASUUME
return
::
*PB12ITI
for IK=1 TO SL
  if SP(IK)=1 THEN PB1=IK: EXIT for
next IK
for IK=SL TO 1 STEP -1
  if SP(IK)=1 THEN PB2=IK: EXIT for
next IK
return
::
*WAKUSYUKUSYO
LVK=1
' case_A
if WKL(STPAN)<EPL(TY,G,STBAN) then SBAN=STPAN:gosub *KUBUNSIROUME: return
if WKL(LTPAN)<EPL(TY,G,LTBAN) then SBAN=LTPAN:gosub *KUBUNSIROUME: return
' case_B( TOTALMINLG>1_case )

```



```

for I9=STPAN+1 to LTPAN-1
  if WKL(I9)<TOTALMINLG then SBAN=I9:gosub *KUBUNSIROUME: return
next I9
return
::
*SPEPIECEIRE
' -- case_A( 2-kubun_made ) --
LVK=0
if (LEVEL<=2)and((NPN=1)or((LEVEL=2)and(NPN=2))) then
  SL=LAST-START+1:SITI=START: gosub *RETUPIECEIRE
  if HSU>=1 THEN FGYOSYO=1: return
end if
' -- case_B( 1-kubun & 2-piece ) --
if (LEVEL>=3)and(NPN=1)and(BMASUN=2)and(BMA(STPAN)=1) then
  SBAN=STPAN:PLG1=EPL(TY,G,STBAN):PLG2=EPL(TY,G,LTBAN): gosub *PIECE2SET
  if HSU>=1 then LVK=3: gosub *EMAKAKIKAE: FGYOSYO=1
end if
return
::
*PIECE2SET
' -- par. SBAN, PLG1,PLG2 --
KBS=SBAN:gosub *MSNARABILIST ' masu-narabi_list
BMLG=B9T+B9-B1T: if (BNSU>=2)and(PLG1<BMLG)and(PLG2<BMLG) then gosub *MASUOKI
return
::
*MASUOKI
' kuromasu_oki
MSTI=1:UMEIT1=B1+B1T:UMEIT2=PLG1: gosub *KUROSIROMASUUME
  UMEIT1=SL-PLG2+1:UMEIT2=B9T-1: gosub *KUROSIROMASUUME
  UMEIT1=B9+B9T:UMEIT2=B1+B1T+PLG2: gosub *KUROSIROMASUUME
  UMEIT1=B9T-PLG1-1:UMEIT2=B1T-1: gosub *KUROSIROMASUUME
' siromasu_ume
MSTI=-1:UMEIT1=1:UMEIT2=B1+B1T-1-PLG1: gosub *KUROSIROMASUUME
  UMEIT1=B9T+PLG2:UMEIT2=SL: gosub *KUROSIROMASUUME
  UMEIT1=B1T+PLG1:UMEIT2=B9+B9T-PLG2-1:gosub *KUROSIROMASUUME
return
::
*KYOTUKUROMASU
LVK=0
if ((LEVEL=2)and(NPN=2)and(BMASUN<=2))or(LEVEL>=3) then
  ' kyotu-kuroMasu
  SL=LAST-START+1:SITI=START: gosub *SPSAKUSEI
  for I9=STBAN TO LTBAN
    MSTI=1:UMEIT1=PITI(I9,2)-EPL(TY,G,I9)-SITI+2:UMEIT2=PITI(I9,1)+EPL(TY,G,I9)-SITI
    gosub *KUROSIROMASUUME
  ' aki-kubun_siromasu
  if I9<LTBAN then MSTI=-1:UMEIT1=PITI(I9,2)-SITI+2:UMEIT2=PITI(I9+1,1)-SITI:gosub *KUROSIRO-
  MASUUME

```

```

next I9
if HSU>=1 then
  if FPITIHEN=1 then LVK=4
  gosub *EMAKAKIKAE: FGYOSYO=1 ' EMA_kakikae
end if
end if
return
::
*LOCALSYORI
for LSYOTY=1 to 7
for CS=1 TO 2
if (LSYOTY=2)or(LSYOTY=3)or(LSYOTY=7) then LVK=2 else LVK=3
' CS=1..left or up, CS=2..right or down
KBS=0:gosub *CSSYORI
HSU=0
if BMA(SPBAN)=1 then
if (LSYOTY=1)or((LEVEL>=2)and(LSYOTY>=2)and(LSYOTY<=6)) then
' Masu-narabi_list
gosub *MASUNARABI
S1=SNLT(1):B1=BNLT(1):S2=SNLT(2):B2=BNLT(2): S9=SNLT(SNSU):B9=BNLT(BNSU)
if LSYOTY<>2 then BIT1=BNIT(1)-1:BMLG=B1 else BIT1=BNIT(BNSU)-1:BMLG=B9
if (LSYOTY>=2)and(LSYOTY<=5) then BG1=GBAN:BG2=HBAN: gosub *PIECEGAITOSU
' syori_wake
select case LSYOTY
case 1 ' kihon_syori
TDSP=SPBAN:TDPB=GBAN:gosub *KIHONSYORI
case 2 ' gyaku-gawa_masu-nobi
if (NPN>=2)and(BNSU<=2)and(S9<=MXKYORI) then gosub *REVSIDEMSNOBI
case 3 ' hanare-masu_syori
if PLG+1<=S1 then gosub *HANAREMASUSYORI1
case 4 ' huku-piece_syori
if (LEVEL>=3)and(S1<=MXKYORI)and(BNSU>=2) then gosub *HUKUSUMASUSYORI
case 5 ' hanare-masu_syori
if (LEVEL>=3)and(PLG+1<=S1) then gosub *HANAREMASUSYORI2
case 6 ' next-kubun_piece-jyoho
if (LEVEL>=3)and(NPN>=2)and(BMASUN>=2) then gosub *nextKUBUNMASU
end select
end if
else
if (LSYOTY=7)and(LEVEL>=2)and(NPN>=2) then gosub *FIRSTBMASU
end if
' EMA_kakikae
if HSU>=1 then gosub *EMAKAKIKAE: FGYOSYO=1: return
next CS
next LSYOTY
return
::
*KIHONSYORI

```

```

' -- par. TDSP, TDPB --
KIHONSU=3: if LEVEL>=2 then KIHONSU=5
for KIHONTY=1 to KIHONSU
  if KIHONTY<=3 then LVK=1 else LVK=2
  select case KIHONTY
    case 1 ' masu-nobi
      if S1<PLG+1 then gosub *MASUNOBI
    case 2 ' masu-narabi_cut
      if (BNSU>=2)AND(S2=1) THEN gosub *MSNARABICUT
    case 3 ' 1piece-narabi_syori
      TDPB=GBAN:gosub *PLG1NARABI ' 1piece_narabi
      if (PLG1SU>0)and(S1>=2) then gosub *PLG1SYORI
    case 4 ' first-piece_ire
      if SL<=2*PLG-1 then PN=1:PRL(1)=PLG: gosub *PIECEIRE
    case 5 ' one-piece_ire
      KNXPB=TDPB+ZGEN:KNXPB=EPL(TY,G,KNXPB)
      if (STBAN<=KNXPB)AND(KNXPB<=LTBAN)AND(SL<=PLG+KNXPB) THEN gosub *PIECE1SET
  end select
  if HSU>=1 THEN return
next KIHONTY
return
::
*PRELOCALSYORI
for CS=1 to 2
  KBS=0:gosub *CSSYORI
  if BMA(SPBAN)=1 then LVK=3: gosub *KUBUNNORIKOSI ' kubun-norikosi
  if BMA(SPBAN)=0 then ' 1piece-narabi_syori
  TDPB=GBAN:gosub *PLG1NARABI: MTPL1=PLG1SU
  if (PLG1SU>0)and(SL<=2*(PLG1SU-1)) then LVK=3: gosub *PLG1SYORI2
  if (FGYOSYO=0)and(LEVEL>=4)and(MTPL1>0) then LVK=4: PLG1SU=MTPL1:gosub *PLG1SYORI3
  end if
  if (FGYOSYO=1)or(FPRELC=1) then return
next CS
return
::
*GLOBALSYORI
LVK=3: gosub *KMASULENSYORI
if (HSU=0)and(LEVEL>=4) then LVK=4: gosub *TOTALMXLG2SYO
if (HSU=0)and(LEVEL>=4) then LVK=4: gosub *NOKUROITI
  if HSU>=1 then gosub *EMAKAKIKAE: FGYOSYO=1 ' EMA_kakikae
if (FGYOSYO=0)and(LEVEL>=4) then LVK=4: gosub *RETUKANSEI2
if (FGYOSYO=0)and(LEVEL>=4) then gosub *PITITIJIME
return
::
*CSSYORI
  if CS=1 then SPBAN=STPAN:LPBAN=LTPAN:GBAN=STBAN:HBAN=LTBAN:ZGEN=1
  if CS=2 then SPBAN=LTPAN:LPBAN=STPAN:GBAN=LTBAN:HBAN=STBAN:ZGEN=-1
  if KBS=0 then KBS=SPBAN

```

```

SL=WKL(KBS):PLG=EPL(TY,G,GBAN)
if CS=1 then SITI=GEN(KBS) else SITI=GEN(KBS)+SL-1
gosub *SPSAKUSEI
return
::
*SPSAKUSEI
' -- par. CS, SL,SITI --
for IK=1 TO SL
if CS=1 THEN SP(IK)=ETI(SITI+IK-1) else SP(IK)=ETI(SITI+1-1K)
next IK
SP(0)=-1: SP(SL+1)=-1
return
::
*MASUNARABI
' -- par. SL,SP() --
FAB=0: SNSU=0:SNLG=0: BNSU=0:BNLG=0
if SP(1)=0 THEN SNITI=1 else SNITI=0
for IK=1 TO SL
if SP(IK)=1 THEN
if FAB=0 then
SNSU=SNSU+1:SNLT(SNSU)=SNLG:SNIT(SNSU)=SNITI
FAB=1: BNLG=1:BNITI=IK
else
BNLG=BNLG+1
end if
else
if FAB=1 then
BNSU=BNSU+1:BNLT(BNSU)=BNLG:BNIT(BNSU)=BNITI
FAB=0: SNLG=1:SNITI=IK
else
SNLG=SNLG+1
end if
end if
next IK
'
if FAB=1 then
BNSU=BNSU+1:BNLT(BNSU)=BNLG:BNIT(BNSU)=BNITI
SNSU=SNSU+1:SNLT(SNSU)=0 :SNIT(SNSU)=0
else
SNSU=SNSU+1:SNLT(SNSU)=SNLG:SNIT(SNSU)=SNITI
end if
return
::
*MSNARABILIST
' -- par. KBS --
SL=WKL(KBS): if CS=1 then SITI=GEN(KBS) else SITI=GEN(KBS)+SL-1
gosub *SPSAKUSEI: gosub *MASUNARABI
S1=SNLT(1):B1=BNLT(1):S2=SNLT(2):B2=BNLT(2): B9=BNLT(BNSU)

```

```

BIT=BNIT(1):B9T=BNIT(BNSU): BIT1=BNIT(1)-1
return
::
*EMAKAKIKAE
' -- par. CS, SL,SITI --
LVKSU(LVK)=LVKSU(LVK)+1
FCHECK=1: CH(TY,G)=1
for IK=1 to SL
  if CS=1 then GITI=SITI+IK-1 else GITI=SITI+1-1K
  if TY=0 then EMA(GITI,G)=SP(IK) else EMA(G,GITI)=SP(IK)
next IK
' CH_kiroku
for IK=1 to HSU
  if CS=1 then GITI=SITI+HLT(IK)-1 else GITI=SITI+1-1K
  CH(1-TY,GITI)=1
next IK
return
::
*PIECEGAITOSU
' -- par. FSIBORI, BG1,BG2, BIT1,BMLG, SL,SITI,ZGEN --
KANOSU=0
' BMIT1,BMIT2_keisan
  if CS=1 then BMIT1=SITI+BIT1:BMIT2=BMIT1+BMLG-1 else BMIT2=SITI-BIT1:BMIT1=BMIT2-
BMLG+1
'
for IK=BG1 TO BG2 STEP ZGEN
  KNHAN=0
  if (EPL(TY,G,IK)>=BMLG)AND(EPL(TY,G,IK)<=SL) THEN
    if LEVEL=1 THEN KNHAN=1
    if LEVEL>=2 THEN
      if BMASUN<=MAXPSU(LEVEL) THEN ' BMASU-sukunai!
        if (PITI(IK,1)<=BMIT1)AND(PITI(IK,2)>=BMIT2) THEN KNHAN=1
        else ' BMASU-oi !
          if (CS=1)AND(PITI(IK,1)<=BMIT1) THEN KNHAN=1
          if (CS=2)AND(PITI(IK,2)>=BMIT2) THEN KNHAN=1
        end if
      end if
    '
    if KNHAN=1 THEN KANOSU=KANOSU+1: KANOLT(KANOSU)=IK
  end if
next IK
' kanosu_sibori
if (FSIBORI=1)AND(KANOSU=2)AND(BNSU>=2) THEN
  if ((LEVEL=3)and(S1=0))or(LEVEL>=4) then
    GT1=0
    for IK=1 TO 2
      PIL=EPL(TY,G,KANOLT(IK)): FGO=1
      for IJ=PIL+1 TO S1+PIL+1

```

```

    if SP(IJ)<>1 then FGO=0
  next IJ
  if (S1<=PIL)and(FGO=1) then GT1=KANOLT(3-IK):exit for
  next IK
  ' hantei
  if GT1>0 THEN KANOSU=1: KANOLT(1)=GT1
end if
end if
' KANOTY & KBG,KPL,HRSU_kime
KBG=KANOLT(1):KPL=EPL(TY,G,KBG):HRSU=abs(KBG-BG1)
gosub *KANOTYPE
return
::
*KANOTYPE
if KANOSU=0 then KANOTY=0: MNKNLG=0: return
gosub *KANOLTMNX
if MNKNLG=MXKNLG then KANOTY=1 else KANOTY=2
return
::
*KANOLTMNX
MNKNLG=1000:MXKNLG=0
for IK=1 to KANOSU
  if EPL(TY,G,KANOLT(IK))<MNKNLG then MNKNLG=EPL(TY,G,KANOLT(IK))
  if EPL(TY,G,KANOLT(IK))>MXKNLG then MXKNLG=EPL(TY,G,KANOLT(IK))
next IK
return
::
*MASUNOBI
GITI=S1+B1+1
if (S1+B1)<=PLG THEN ' case_A
  MSTI=1:UMEIT1=GITI:UMEIT2=PLG: gosub *KUROSIROMASUUME
  if (S1=0)AND(SP(PLG+1)=0) THEN SP(PLG+1)=-1:HSU=HSU+1: HLT(HSU)=PLG+1
else ' case_B
  MSTI=-1:UMEIT1=1:UMEIT2=S1+B1-PLG: gosub *KUROSIROMASUUME
  if (B1=PLG)AND(SP(GITI)=0) THEN SP(GITI)=-1:HSU=HSU+1: HLT(HSU)=GITI
end if
return
::
*MSNARABICUT
' -- par. TDPB --
BMLG=B1+B2+1: FGO=0
if (S1<PLG+1)and(BMLG>PLG) then FGO=1 ' case_A
if (S1>=PLG+1)and(S1<=MXKYORI) then ' case_B
  BG1=TDPB:BG2=HBAN: gosub *PIECEGAITOSU
  if KANOSU=0 THEN FGO=1
end if
'
if FGO=1 THEN SP(SNIT(2))=-1: HSU=1:HLT(1)=SNIT(2)

```

```

return
::
*PLG1SYORI
if PLG1SU>=MAXPSU(LEVEL) THEN PLG1SU=MAXPSU(LEVEL)-1
if (PLG1SU=1)AND(S1=2) THEN ' case_A
  SP(2)=-1: HSU=1:HLT(1)=2
  if B1>1 THEN SP(1)=1: HSU=2:HLT(2)=1
else ' case_B
  if (PLG1SU>=2)AND(S1<2*PLG1SU+1) THEN
    SP(BIT1)=-1: HSU=HSU+1:HLT(HSU)=BIT1
    if (B1=1)AND(S1<2*PLG1SU) THEN SP(BIT1+2)=-1: HSU=HSU+1:HLT(HSU)=BIT1+2
  end if
end if
return
::
*PLG1NARABI
' -- par. TDPB, ZGEN --
PLG1SU=0: NXLG2=0
for IK=TDPB TO HBAN STEP ZGEN
  if EPL(TY,G,IK)=1 THEN PLG1SU=PLG1SU+1 else NXLG2=EPL(TY,G,IK):EXIT for
next IK
return
::
*REVSIDEMSNOBI
' itti_nagasa
if (KANOTY=1)and(MNKNLG=B9) then W1=BIT1:W2=BIT1+B9+1: gosub *W1W2SIRO
' masu_nobi
if MNKNLG>S9+B9 then MSTI=1:UMEIT1=SL-MKNLG+1:UMEIT2=BIT1:gosub *KUROSIROMASU-
UME
if HSU>=1 then return
' space_ume
if (LEVEL>=3)and(KANOTY=1)and(KANOSU<=MAXPSU(LEVEL-1)) then
  NEXMIN=1000 ' NEXMIN_keisan
  for IK=1 to KANOSU
    NEXB=KANOLT(IK)+ZGEN
    if (STBAN<=NEXB)and(NEXB<=LTBAN)and(EPL(TY,G,NEXB)<NEXMIN) then
      NEXMIN=EPL(TY,G,NEXB)
  next IK
'
  if S9<=NEXMIN then LVK=3: MSTI=-1:UMEIT1=BIT1+MNKNLG+1:UMEIT2=SL:gosub *KUROSIRO-
MASUUME
end if
return
::
*W1W2SIRO
if SP(W1)=0 then SP(W1)=-1:HSU=HSU+1:HLT(HSU)=W1
if SP(W2)=0 then SP(W2)=-1:HSU=HSU+1:HLT(HSU)=W2
return

```

```

::
*HANAREMASUSYORI1
' case_A
if KANOTY=1 then
if (KANOSU=1)and(HRSU<=MXHANARE(LEVEL)) then
' masu_nobi & siroM_ume
if (HRSU=0)and((S1<=MXKYORI)or(B1>TOTALMXLG2)) then
MSTI=-1:UMEIT1=1:UMEIT2=S1+B1-PLG: gosub *KUROSIROMASUUME: return
end if
if (LEVEL>=3)and(S1<=MXKYORI)and(HRSU>=1) then
PLWA=-1: for I9=GBAN to KBG step ZGEN:PLWA=PLWA+EPL(TY,G,I9)+1: next I9
if PLWA>S1+B1 then LVK=3: MSTI=1:UMEIT1=S1+B1+1:UMEIT2=PLWA:gosub *KUROSIROMASU-
UME: return
end if
end if
' itti_nagasa
if MNKNLG=B1 then W1=BIT1:W2=BIT1+B1+1:gosub *W1W2SIRO: return
end if
' case_B
if (LEVEL>=3)and(S1<=MXKYORI)and(PLG=B1)and(PLG>1) then
if PLG=S1-1 then LVK=3: SP(BIT1)=-1:HSU=HSU+1:HLT(HSU)=BIT1: return
if LEVEL>=4 then
MBIT=BIT1: BIT1=BIT1-1:BMLG=B1+1:gosub *PIECEGAITOSU
FGO=0: if KANOSU=0 then FGO=1
if KANOSU>0 then
PLWA=-1: for I9=GBAN to KBG-ZGEN step ZGEN:PLWA=PLWA+EPL(TY,G,I9)+1: next I9
if PLWA>S1-2 then FGO=1
if FGO=1 then LVK=4: SP(MBIT)=-1:HSU=HSU+1:HLT(HSU)=MBIT
end if
end if
end if
return
::
*HANAREMASUSYORI2
if (KANOSU=1)and(HRSU>=1)and(HRSU<=MXHANARE(LEVEL)) then
' piece_ire
PN=HRSU: for I9=1 to PN: PRL(I9)=EPL(TY,G,GBAN+(I9-1)*ZGEN): next I9
SL=S1-1: gosub *PIECEIRE
end if
return
::
*HUKUSUMASUSYORI
if (KANOSU=1)AND(HRSU<=MXHANARE(LEVEL)) THEN
ABAN=KBG:APLG=KPL
S2T=SNIT(2):BIT2=BNIT(2)-1: BG1=ABAN+ZGEN:BG2=HBAN:TGPL=EPL(TY,G,BG1)
if S2=1 then ' case_A
if (BNLT(2)>TGPL)or(SP(S2T+TGPL+1)=1) then SP(S2T)=1:HSU=1:HLT(1)=S2T: return
else

```





```

return
::
*NEXTKUBUNMASU
NXPBAN=SPBAN+ZGEN: NXBAN=GBAN+ZGEN
if BMA(NXPBAN)=1 then
  KBS=NXPBAN:gobsub *MSNARABILIST ' masu-narabi_list
  BG1=NXBAN:BG2=HBAN:BMLG=B1: FSIBORI=1:gobsub *PIECEGAITOSU:FSIBORI=0
  if (KANOSU=1)and(S1<=MXKYORI)and(abs(GBAN-KBG)<=MXHANARE(LEVEL)) then
    '- case_A( 1piece-ire ) -
    if HRSU=0 then MKSYOTY=1:gobsub *MKUBUNSYO: return
    '- case_B( moto-kubun_space-ire ) -
    PNMIN=2:gobsub *MOTOSPACEIRE
  end if
end if
return
::
*FIRSTBMASU
BMSP=0 ' first_blockMASU
for I9=SPBAN to LPBAN step ZGEN
  if BMA(I9)=1 then BMSP=I9: exit for
next I9
if (BMSP=0)or(abs(BMSP-SPBAN)>MXHANARE(LEVEL)) then return
' SL,SP_sakusei & MASU-narabi
KBS=BMSP:gobsub *MSNARABILIST ' masu-narabi_list
BG1=GBAN:BG2=HBAN:BMLG=B1: FSIBORI=1:gobsub *PIECEGAITOSU:FSIBORI=0
if (KANOSU=1)and(S1<=MXKYORI) then
  '- case_A( siro-masu_ume ) -
  if HRSU=0 then MKSYOTY=2:gobsub *MKUBUNSYO: return
  '- case_B( moto-kubun_space-ire ) -
  if (LEVEL>=3)and(abs(BMSP-SPBAN)=1)and(HRSU<=MXHANARE(LEVEL)) then
    LVK=3: PNMIN=1:gobsub *MOTOSPACEIRE
  end if
end if
return
::
*MKUBUNSYO
' -- par. MKSYOTY --
SL=WKL(SPBAN): if CS=1 then SITI=START else SITI=LAST
gobsub *SPSAKUSEI
if MKSYOTY=1 then PLG=EPL(TY,G,GBAN):gobsub *PIECE1SET
if MKSYOTY=2 then MSTI=-1:UMEIT1=1:UMEIT2=SL:gobsub *KUROSIROMASUUME
return
::
*MOTOSPACEIRE
' -- par. PNMIN,S1 --
TBML=0:PN=0
for J9=KBG-ZGEN TO GBAN STEP -ZGEN
  TBML=TBML+EPL(TY,G,J9)+1

```

```

    if TBML>S1 THEN PN=ABS(J9-GBAN+ZGEN): EXIT for
  next J9
  if PN>=PNMIN then
    SL=WKL(SPBAN): if CS=1 then SITI=START else SITI=LAST
    for J9=1 TO PN: PRL(J9)=EPL(TY,G,GBAN+(J9-1)*ZGEN): next J9
    gosub *SPSAKUSEI: gosub *PIECEIRE
  end if
  return
::
*KUBUNNORIKOSI
' -- norikosi_jyunbi --
NXPB=GBAN+ZGEN:NXPLG=EPL(TY,G,NXPB): NX2PB=NXPB+ZGEN
NXSP=SPBAN+ZGEN:NSPL=WKL(NXSP)
if SL<=(PLG+NXPLG) then
  NKTY=1:FPRELC=1: gosub *PIECE1SET      ' piece-1set
else
  if (STBAN<=NX2PB)and(NX2PB<=LTBAN) then
    NX2PLG=EPL(TY,G,NX2PB)
    if SL<=PLG+NXPLG+NX2PLG+1 then
      if ((BMA(NXSP)=1)and(NSPL<NX2PLG))or((TML(NXSP)=1)and(NSPL<>NX2PLG)) then
        NKTY=1:FPRELC=1:gosub *PIECE1SET      ' piece-1set
      end if
    ' piece-2set
    gosub *MASUNARABI
    B1=BNLT(1):B1T=BNIT(1):B9=BNLT(BNSU):B9T=BNIT(BNSU): BMLG=B9T+B9-B1T
    if (BNSU>=2)and(BMLG>PLG)and(BMLG>NXPLG) then
      NKTY=2:FPRELC=1      ' piece-2set
      PLG1=PLG:PLG2=NXPLG: gosub *MASUOKI
    end if
  end if
end if
end if
end if
if HSU>=1 then gosub *EMAKAKIKAE:FGYOSYO=1: FPRELC=0
' -- sagyo-retu & piece_syukusyo --
if FPRELC=1 then
  if CS=1 then STPAN=STPAN+1:STBAN=STBAN+NKTY else LTPAN=LTPAN-1:LTBAN=LTBAN-NKTY
  NPN=NPN-1:BMA SUN=BMA SUN-NKTY: START=GEN(STPAN):LAST=GEN(LTPAN)+WKL(LTPAN)-1
end if
return
::
*PLG1SYORI2
if PLG1SU>=MAXPSU(LEVEL) THEN PLG1SU=MAXPSU(LEVEL)-1
' 1-masu_narabi
MASU1N=0
for I9=SPBAN TO LPBAN STEP ZGEN
  if BMA(I9)=0 THEN
    MASU1N=MASU1N+INT((WKL(I9)+1)/2)
  else

```

```

if WKL(I9)=1 THEN
  MASU1N=MASU1N+1
else
  KBS=I9:gosub *MSNARABILIST ' masu-narabi_list
  ' case_A
  NOKORI1=PLG1SU-MASU1N
  if (B1=1)AND(S1<2*NOKORI1+1) THEN
    if SP(BIT1)=0 THEN SP(BIT1)=-1: HSU=HSU+1:HLT(HSU)=BIT1
    if (S1<2*NOKORI1)AND(SP(BIT1+2)=0) THEN SP(BIT1+2)=-1: HSU=HSU+1:HLT(HSU)=BIT1+2
    if HSU>=1 THEN gosub *EMAKAKIKAE:FGYOSYO=1: return
  end if
  ' case_B
  if (MTPL1=PLG1SU)AND(B1>=2)AND(MASU1N+INT(S1/2)=PLG1SU) THEN
    for I8=SPBAN TO I9 STEP ZGEN
      if (BMA(I8)=0)or(I8=I9) THEN
        if I8<>I9 THEN SL=WKL(I8) else SL=S1-1
        if CS=1 THEN SITI=GEN(I8) else SITI=GEN(I8)+WKL(I8)-1
        gosub *SPSAKUSEI
        PN=INT((SL+1)/2): for IJ=1 TO PN: PRL(IJ)=1: next IJ
        HSU=0:gosub *PIECEIRE
        if HSU>=1 THEN gosub *EMAKAKIKAE:FGYOSYO=1
        end if
      next I8
    end if
    return
  end if
end if
'
if MASU1N>PLG1SU THEN EXIT for
next I9
return
::
*PLG1SYORI3
if (PLG1SU<MAXPSU(LEVEL))and(NXLG2>0)and(WKL(SPBAN)<2*PLG1SU+NXLG2) then
  MASU1N=0: ' 1-masu_narabi-su
  for I9=SPBAN+ZGEN TO LPBAN STEP ZGEN
    if WKL(I9)>=NXLG2 THEN return
  ' MASU1N_keisan
  if BMA(I9)=1 THEN
    SL=WKL(I9): if CS=1 then SITI=GEN(I9) else SITI=GEN(I9)+SL-1
    gosub *SPSAKUSEI: gosub *MASUNARABI
    MASU1N=MASU1N+BNSU
  end if
  ' hantei( siro-masu_ume )
  if MASU1N=PLG1SU then
    MKSYOTY=2:gosub *MKUBUNSYO
    gosub *EMAKAKIKAE:FGYOSYO=1: return
  end if

```

```

next I9
end if
return
::
*KMASULENSYORI
for I9=STPAN TO LTPAN
' case_A
if (BMA(I9)=1)AND(TML(I9)=0) THEN
SL=WKL(I9)
for SYOTY=1 to 4
if SYOTY=1 then CSN=2 else CSN=1
for CS=1 to CSN
LVK=3
KBS=I9:gosub *CSSYORI: gosub *MASUNARABI
' hantei
select case SYOTY
case 1 ' masu-nobi_from-haji
S1=SNLT(1):B1=BNLT(1): BIT1=BNIT(1)-1: RBMIT=S1+B1
BG1=GBAN:BG2=HBAN:BMLG=B1: gosub *PIECEGAITOSU
if RBMIT<MNKNLG then
MSTI=1:UMEIT1=RBMIT+1:UMEIT2=MNKNLG: gosub *KUROSIROMASUUME
if (KANOTY=1)and(S1=0)and(SP(MNKNLG+1)=0) then SP(MNKNLG+1)=-1:
HSU=HSU+1:HLT(HSU)=MNKNLG+1
end if
case 2 ' narabi_MAX-syori
for I8=1 to BNSU
B0=BNLT(I8): W1=BNIT(I8)-1:W2=BNIT(I8)+BNLT(I8)
BG1=STBAN:BG2=LTBAN:BMLG=B0:BIT1=W1: gosub *PIECEGAITOSU
if (KANOTY=1)and(B0=MXKNLG) then gosub *W1W2SIRO
next I8
case 3 ' laki-narabi
for I8=2 to BNSU
if SNLT(I8)=1 THEN
B1=BNLT(I8-1):B2=BNLT(I8):SIT1=SNIT(I8):BIT1=BNIT(I8-1)-1
BG1=STBAN:BG2=LTBAN: BMLG=B1+B2+1: gosub *PIECEGAITOSU
if KANOSU=0 then SP(SIT1)=-1:HSU=HSU+1:HLT(HSU)=SIT1 ' cut
if (KANOSU>0)and(LEVEL>=4) then ' kuroMS_ume
LVK=4
for I7=1 to 2
if I7=1 then BMLG=B1:HKLG=B2:ZPB=1 else BMLG=B2:HKLG=B1:ZPB=-1:BIT1=BNIT(I8)-1
gosub *PIECEGAITOSU
if KANOSU=1 then
NXBG=KBG+ZPB:NXP=LEPL(TY,G,NXBG):SIT2=SIT1+(NXP+1)*ZPB
if HKLG>NXP then SP(SIT1)=1:HSU=HSU+1:HLT(HSU)=SIT1
if (1<=SIT2)and(SIT2<=SL) then
if SP(SIT2)=1 then SP(SIT1)=1:HSU=HSU+1:HLT(HSU)=SIT1
end if
end if

```

```

    if HSU>=1 then exit for
    next I7
end if
end if
next I8
case 4 ' block_rihan
if (LEVEL>=4)and(BNSU>=2) then
LVK=4
for I8=1 to BNSU
BG1=STBAN:BG2=LTBAN:BMLG=BNLT(I8):BIT1=BNIT(I8)-1: gosub *PIECEGAITOSU
if KANOSU=0 then MX(I8)=1000:MN(I8)=0 else MX(I8)=MXKNLG:MN(I8)=MNKNLG
next I8
for I8=1 to BNSU
B0=BNLT(I8-1):S1=SNLT(I8):B1=BNLT(I8):S2=SNLT(I8+1):B2=BNLT(I8+1)
S1T=SNIT(I8):B1T=BNIT(I8):B2T=BNIT(I8+1): MSTI=1
if (I8<BNSU)and(B1+S2+B2>MX(I8)) then UMEIT1=B2T-MN(I8):UMEIT2=B1T-1:gosub
*KUROSIROMASUUME
if (I8>1)and(B0+S1+B1>MX(I8-1)) then UMEIT1=B1T+B1:UMEIT2=S1T+MN(I8):gosub
*KUROSIROMASUUME
next I8
end if
end select
if HSU>=1 then return
next CS,SYOTY
end if
' case_B
if (LEVEL>=4)and(BMA(I9)=0)and(WKL(I9)<TOTALMXLG) then
CS=1: SL=WKL(I9):SITI=GEN(I9):ZGEN=1: gosub *SPSAKUSEI
BG1=STBAN:BG2=LTBAN:BIT1=0:BMLG=1: gosub *PIECEGAITOSU
if KANOSU=0 then LVK=4: MSTI=-1:UMEIT1=1:UMEIT2=SL:gosub *KUROSIROMASUUME: return
end if
next I9
return
::
*TOTALMXLG2SYO
if (TOTALMXLG>1)and(TOTALMXLG2=1) then
SL=LAST-START+1:SITI=START:CS=1: gosub *SPSAKUSEI
gosub *BMRENSIRABE ' 2-BMblock_sirabe
if FPLG2=1 THEN
for IK=1 TO SL
if ((SP(IK-1)=-1)OR(SP(IK-1)=0))AND(SP(IK)=1)AND((SP(IK+1)=-1)OR(SP(IK+1)=0)) THEN
FGO=0 ' setuzoku_hantei
if IK<BMRENST THEN SETULG=BMRENST+BMREN-1 else SETULG=IK-BMRENST+1
if SETULG>TOTALMXLG THEN
FGO=1
else
if IK<BMRENST THEN IST=IK:ILT=BMRENST else IST=BMRENST+BMREN:ILT=IK
for I8=IST TO ILT

```

```

    if SP(I8)=-1 THEN FGO=1: EXIT for
  next I8
end if
,
  if FGO=1 THEN W1=IK-1:W2=IK+1:gosub *W1W2SIRO
end if
next IK
end if
end if
return
::
*BMRENSIRABE
FBMREN=0:BMREN=0: FPLG2=0
for IK=1 TO SL
  if SP(IK)=1 THEN
    if FBMREN=0 THEN
      FBMREN=1:BMREN=1
    else
      BMREN=BMREN+1: if BMREN=2 THEN BMRENST=IK-1
    end if
  else
    if BMREN>=2 THEN FPLG2=1: return
    ' modosi
    FBMREN=0:BMREN=0
  end if
next IK
if BMREN>=2 THEN FPLG2=1
return
::
*NOKUROITI
for CS=1 to 2
  if CS=1 then PKB=STBAN:HBAN=LTBAN:SPBAN=STPAN:LPBAN=LTPAN: ZGEN=1
  if CS=2 then PKB=LTBAN:HBAN=STBAN:SPBAN=LTPAN:LPBAN=STPAN: ZGEN=-1
  for I9=SPBAN to LPBAN step ZGEN
    if BMA(I9)=1 then
      KBS=I9:gosub *MSNARABILIST ' masu-narabi_list
      if TML(I9)=1 then
        BG1=PKB:BG2=HBAN:BIT1=0:BMLG=SL:gosub *PIECEGAITOSU
        if KANOSU=1 then PKB=KBG+ZGEN else exit for
      else
        if (BNSU=1)and(S2=0) then
          BG1=PKB:BG2=HBAN:BMLG=B1:gosub *PIECEGAITOSU
          KBG2=KANOLT(2):KPL2=EPL(TY,G,KBG2)
          if (KANOSU=2)and(KBG=PKB)and(KBG2=KBG+ZGEN)and(KPL<KPL2) then
            if SP(SL-KPL2)=0 then SP(SL-KPL2)=-1: HSU=1:HLT(1)=SL-KPL2
          end if
        end if
      end if
    exit for
  next I9
end if

```

```

    end if
  end if
next I9
if HSU>=1 then return
next CS
return
::
*RETUKANSEI2
  gosub *RTKANHAN
  if FRTKAN=1 then gosub *KUBUNSYORI ' kubun_syori
  return
::
*RTKANHAN
  FRTKAN=0
  CS=1: gosub *KBLOCKSU
  if KBLSU=BMASUN then FRTKAN=1:BRKBN=1000: return
  '
  if KBLSU=BMASUN-1 then
    for I9=1 to KBLSU
      KBS=BMAL(I9):gosub *MSNARABILIST ' masu-narabi_list
      if BNSU>=2 then
        PLG=B9T+B9-B1T:GPICN1=EPL(TY,G,STBAN+I9-1):GPICN2=EPL(TY,G,STBAN+I9)
        if (PLG>GPICN1)and(PLG>GPICN2) then FRTKAN=1:BRKBN=I9: exit for
      end if
    next I9
  end if
  return
::
*PITITIJIME
  for I9=STPAN TO LTPAN
    if BMA(I9)=1 THEN
      SL=WKL(I9):SITI=GEN(I9): ZGEN=1
      CS=1:gosub *SPSAKUSEI: gosub *MASUNARABI
      '
      BG1=STBAN:BG2=LTBAN
      for I8=1 TO BNSU
        BIT1=BNIT(I8)-1:B1=BNLT(I8):BMLG=B1:gosub *PIECEGAITOSU
        if KANOSU=0 then return
        ' hantei
        KXBG=KANOLT(KANOSU)
        HAN1=BNIT(I8)+B1-EPL(TY,G,KXBG): if HAN1<1 then HAN1=1
        HAN2=BIT1+KPL:          if HAN2>SL then HAN2=SL
        HANSU1=HAN1+SITI-1:HANSU2=HAN2+SITI-1
        if HANSU1>PITI(KXBG,1) THEN
          FPITIHEN=1:PITI(KXBG,1)=HANSU1
          ' hoka-piece_henka
          ZKAKUITI=HANSU1+EPL(TY,G,KXBG)+1:ZKAKUSB=I9:MBAN=KXBG+1:gosub *PITII
        end if
      next I8
    end if
  next I9

```



```

if HANSU2<PITI(KBG,2) THEN
  FPITIHEN=1:PITI(KBG,2)=HANSU2
  ' hoka-piece_henka
  ZKAKUITI=HANSU2-EPL(TY,G,KBG)-1:ZKAKUSB=I9:MBAN=KBG-1:gosub *PITI2
end if
if FPITIHEN=1 then return
next I8
end if
next I9
return
::
*MESYORI
LVK=3
KHLLC=3:KHLV3TY=2: KHN=0:gosub *KHL34LIST
for I=1 to 8
  KTY=KHL(I,1):KG=KHL(I,2):KSITI=KHL(I,3):RETUN=KHL(I,4):KCS=KHL(I,5):DR=KHL(I,6)
  if RETUN>=2 then
    for I1=1 to RETUN
      if KCS=1 then RIT(I1)=KSITI+(I1-1) else RIT(I1)=KSITI-RETUN+I1
    next I1
    FMESY=1:gosub *KPLTSAKUSEI: FMESY=0      ' KPLT()-sakusei
  '
  for DI=1 to 2
    gosub *BSULSAKUSEI      ' BSUL()_sakusei
  ' hantei
  TY=KTY:G=KG+DR*DI: gosub *ETISAKUSEI:gosub *WAKUKIME
  if KASE(TY,G)=0 then
    FGO=0
    for I1=1 to BSU
      HTY=BSUL(I1,0):BLG=BSUL(I1,1):POTI=BSUL(I1,2)
      if (START>POTI)or(POTI>LAST) then exit for      ' jyogai
      FGOO=1
      for BI=STBAN to LTBAN
        if ((HTY=0)and(BLG<=EPL(TY,G,BI)))or((HTY=1)and(BLG=EPL(TY,G,BI))) then FGOO=0: exit for
      next BI
      if FGOO=1 then FGO=1: exit for
    next I1
    if FGO=1 then
      LVKSU(LVK)=LVKSU(LVK)+1
      if KTY=0 then EMA(KSITI,KG)=-1 else EMA(KG,KSITI)=-1
      CH(KTY,KG)=1:CH(1-KTY,KSITI)=1
      FTYHUKUGO=1: return
    end if
  end if
end if
next DI
end if
next I
return

```

```

::
*TAYOHUKUGOSYO
LVK=4
for KTY=0 to 1:for KG=1 to GR(KTY)
  if KASE(KTY,KG)=0 then
    TY=KTY:G=KG: gosub *ETISAKUSEI:gosub *WAKUKIME
    gosub *PIECEITI: gosub *PITITIJIME
    ' hikae_toru
    MSTPAN=STPAN:MLTPAN=LTPAN:MNPN=NPN: MSTBAN=STBAN:MLTBAN=LTBAN
    for I=STPAN to LTPAN: MWKL(I)=WKL(I):MGEN(I)=GEN(I):MBMA(I)=BMA(I):MTML(I)=TML(I):
next I
  for J=STBAN to LTBAN:for J2=1 to 2: MPITI(J,J2)=PITI(J,J2): next J2,J
  '
  for I=MSTPAN to MLTPAN
    MSL=MWKL(I):MSITI=MGEN(I): TZ=1-KTY
    for HKGTY=1 to 3
      gosub *SPJYUNBI
      for J=MSTBAN to MLTBAN:for J2=1 to 2: PITI(J,J2)=MPITI(J,J2): next J2,J
      if (HKGTY<=2)and(MBMA(I)=1)and(MTML(I)=0) then
        gosub *MASUNARABI
        S1=SNLT(1):B1=BNLT(1):S2=SNLT(2)
        if (BNSU=1)and(S1=S2) then
          BG1=MSTBAN:BG2=MLTBAN:BIT1=BNIT(1)-1:BMLG=B1:gosub *PIECEGAITOSU
          if (KANOTY=1)and(KPL=B1+S1) then
            if HKGTY=1 then gosub *OVERTUMIAGE else gosub *KUBUNTIJIME ' syori_1,2
          end if
        end if
      end if
      if (HKGTY=3)and(MBMA(I)=0)and(MSL>=2) then gosub *AKIKUBUNKETU ' syori_3
      if FTYHUKUGO=1 then return
    next HKGTY
  next I
end if
next KG,KTY
return
::
*SPJYUNBI
TY=KTY:G=KG: CS=1:SL=MSL:SITI=MSITI: ZGEN=1
gosub *ETISAKUSEI:gosub *SPSAKUSEI
return
::
*OVERTUMIAGE
if S1=1 then
  APL=KPL: RIT(1)=SITI:RIT(2)=SITI+SL-1
  for I1=1 to 2:for I2=-1 to 1 step 2
    RT1=RIT(I1):RT2=RIT(3-I1)
    if KTY=0 then LTI=EMA(RT1,KG+I2) else LTI=EMA(KG+I2,RT1)
    if LTI=-1 then

```

```

TY=TZ:G=RT1: DR=-I2
FSU=0: if I2=-1 then MCS=1 else MCS=2
CHTY=1: POTI=KG:MBMLG=1:gosub *PIECEJYOHO
if MNKNLG>=2 then FSU=FSU+1: KPL1=MNKNLG
CHTY=3: G=RT2:POTI=KG:gosub *PIECEJYOHO
if KANOSU>0 then FSU=FSU+1: ALG=MXKNLG-BMLG+2
if (FSU=2)and(ALG<=KPL1) then
  FGO=1
  for J=1 to ALG-1
    TY=KTY:G=KG+DR*J
    if TY=0 then LTI1=EMA(RT1,G):LTI2=EMA(RT2,G) else LTI1=EMA(G,RT1):LTI2=EMA(G,RT2)
    if (LTI1<>0)or(LTI2=-1) then FGO=0: exit for ' check_1
    if LTI2=0 then
      if I1=1 then MCS=1:ZHB=1 else MCS=2:ZHB=-1
      FKR=0:MBMLG=0
      for IJ=1 to APL
        R=RT2-IJ*ZHB: if TY=0 then LTI=EMA(R,G) else LTI=EMA(G,R)
        if FKR=0 then
          if LTI=-1 then FGO=0: exit for
          if LTI=1 then FKR=1:MBMLG=1
          else
            if LTI=1 then MBMLG=MBMLG+1 else POTI=R+ZHB:exit for
          end if
        next IJ
      if (FGO=0)or(MBMLG=0) then FGO=0: exit for ' check_2
      if TY=0 then EMA(RT1,G)=-1 else EMA(G,RT1)=-1 ' karioki
      CHTY=2: BT0=POTI-ZHB:gosub *PIECEJYOHO
      if TY=0 then EMA(RT1,G)=0 else EMA(G,RT1)=0 ' modosi
      if I1=1 then TB=SITI+MNKNLG-1 else TB=SITI-MNKNLG+1
      if ((I1=1)and(TB<RT2))or((I1=2)and(TB>RT2)) then FGO=0: exit for ' check_3
    end if
  next J
  ' hantei
  if FGO=1 then
    LVKSU(LVK)=LVKSU(LVK)+1
    G=KG+DR*(ALG-1)
    if KTY=0 then EMA(RT1,G)=1 else EMA(G,RT1)=1
    CH(KTY,G)=1:CH(1-KTY,RT1)=1
    FTYHUKUGO=1: return
  end if
end if
end if
next I2,I1
else ' S1>=2
RETUN=S1: APL=KPL
for I1=1 to 2
  if I1=1 then STTI=MSITI-1 else STTI=MSITI+APL-1
  for I3=1 to RETUN: RIT(I3)=STTI+I3: next I3

```

```

' -- case_A --
OVTDSYOTY=2: gosub *OVERTOBIDASICHA
if FTYHUKUGO=1 then return
' -- case_B --
if I1=1 then
  OVTDSYOTY=2: gosub *OVERTOBIDASICHB
  if FTYHUKUGO=1 then return
end if
next I1
end if
return
::
*AKIKUBUNKETU
BG1=MSTBAN:BG2=MLTBAN:BIT1=0:BMLG=1:gosub *PIECEGAITOSU
if 2*MNKNLG>MSL then
  MKPL=MNKNLG
  ' -- case_A( siroMASU_ketu ) --
  KRLG=2*MKPL-MSL:KRSTI=MSL-MKPL
  if KRLG>=2 then
    for I1=1 to KRLG: RIT(I1)=KRSTI+I1+MSITI-1: next I1
    RETUN=KRLG:OVTDSYOTY=1: gosub *OVERTOBIDASICHA
    if FTYHUKUGO=1 then return
  end if
  ' -- case_B( kuroMASU_ketu ) --
  for I1=1 to MSL: RIT(I1)=I1+MSITI-1: next I1
  RETUN=MSL:OVTDSYOTY=1: gosub *OVERTOBIDASICHB
end if
return
::
*PIECEJYOHU
' -- par. MCS,POTI, (DR,MBMLG,BTO,MPOTI, FMESY) --
gosub *ETISAKUSEI:gosub *WAKUKIME:gosub *PIECEITI
if LEVEL=5 then gosub *PITITIJIME
gosub *KUBUNSAGASI
CS=MCS: SL=WKL(KBSU)
if CS=1 then SITI=GEN(KBSU):BG1=STBAN:BG2=LTBAN:ZGEN=1
if CS=2 then SITI=GEN(KBSU)+SL-1:BG1=LTBAN:BG2=STBAN:ZGEN=-1
' waku-hazure_jyogai
if (START>POTI)or(POTI>LAST) then
  if FMESY=1 then MNKNLG=SL else KANOSU=-1:KANOTY=-1:MNKNLG=0
  return
end if
' KANOSU_keisan(CHTY_wake)
if CHTY=1 then BIT1=0:BMLG=MBMLG:gosub *PIECEGAITOSU
if CHTY>=2 then
  if CHTY=3 then MBMLG=1:ZU=-DR:gosub *BIT1KIME
  if CHTY=4 then
    POTI=MPOTI+MXPL-1:MBMLG=MXPL:ZU=1: gosub *BIT1KIME

```

```

    POTI=MPOTI:ZU=-1: gsub *BIT1KIME
end if
if CHTY>=3 then BT0=REPIT
' BIT1_kime
if CS=1 then BIT1=BT0-SITI+1 else BIT1=SITI-BT0+1
BMLG=MBMLG:gosub *PIECEGAITOSU
end if
return
::
*BIT1KIME
' -- par. POTI,MBMLG,ZU --
REPIT=POTI+ZU: if TY=0 then RELTI=EMA(REPIT,G) else RELTI=EMA(G,REPIT)
while RELTI=1
    REPIT=REPIT+ZU:MBMLG=MBMLG+1: if TY=0 then RELTI=EMA(REPIT,G) else
RELTi=EMA(G,REPIT)
    wend
return
::
*KUBUNSAGASI
' -- par. POTI --
KBSU=0
for IK=1 to WKN
    if (GEN(IK)<=POTI)and(POTI<=GEN(IK)+WKL(IK)-1) then KBSU=IK:exit for
next IK
return
::
*OVERTOBIDASICHA
' -- par. RETUN,OVTDSYOTY, RIT() --
TYAB=1
for I2=-1 to 1 step 2
    DR=-I2: gsub *KPLTSAKUSEI ' KPLT,MNSYO_sakusei
' case_A1( MXPL_riyo )
    gsub *MXPLRIYO
    if FTYHUKUGO=1 then return
' case_A2( MNPL_riyo )
    MXDN=0
    for I3=1 to RETUN
        if (MNSYO(I3)=1)and(MXDN<KPLT(I3)) then MXDN=KPLT(I3)
    next I3
    for DI=1 to MXDN
        gsub *BSULSAKUSEI ' BSUL()_sakusei
        ' hantei
        TY=KTY: G=KG+DI*DR: SKRHN=0
        if KASE(TY,G)=0 then
            ' karioki
            for I3=1 to SIKIRIN
                if TY=0 then XP=SKRL(I3):YP=G else XP=G:YP=SKRL(I3)
                if EMA(XP,YP)=0 then EMA(XP,YP)=-1: SKRHN=SKRHN+1:MX(SKRHN)=XP:MN(SKRHN)=YP
            next I3
        next DI
    next I2

```

```

next I3
for I4=1 to BSU
MCS=1: POTI=BSUL(I4,2):MBMLG=BSUL(I4,1)
if I4=1 then ZU=-1:gosub *BIT1KIME:BT0=REPIT else BT0=POTI-1
CHTY=2: gosub *PIECEJYOHO
if KANOSU=0 then
for I3=1 to SKRHN: EMA(MX(I3),MN(I3))=0: next I3 ' modosi
gosub *OVTDMSTI: return
end if
next I4
for I3=1 to SKRHN: EMA(MX(I3),MN(I3))=0: next I3 ' modosi
end if
next DI
next I2
return
::
*KPLTSAKUSEI
for I3=1 to RETUN
TY=1-KTY:G=RIT(I3): if DR=1 then MCS=1 else MCS=2
CHTY=3: POTI=KG:gosub *PIECEJYOHO
if MCS=1 then YMSA=MNKNLG+SITI-POTI else YMSA=POTI-(SITI-MNKNLG)
if YMSA>=2 then KPLT(I3)=YMSA else KPLT(I3)=1
if ((BIT1=0)and(KANOTY=1))or(FMESY=1) then MNSYO(I3)=1 else MNSYO(I3)=0
next I3
return
::
*BSULSAKUSEI
BSU=0: FBM=0:BLG=0:FBREN=0: FSKRTY=0:SKRITI=0:SIKIRIN=0
for I3=1 to RETUN
if (MNSYO(I3)=1)and(KPLT(I3)=DI) then
if FBM=1 then
BSU=BSU+1: BSUL(BSU,1)=BLG:BSUL(BSU,2)=BITI
if (FSKRTY=1)and(BLG=RIT(I3)-SKRITI-1) then BSUL(BSU,0)=1 else BSUL(BSU,0)=0
end if
FBM=0:BLG=0:FBREN=0: FSKRTY=1:SKRITI=RIT(I3)
SIKIRIN=SIKIRIN+1: SKRL(SIKIRIN)=SKRITI
else
if KPLT(I3)>DI then
if FBREN=1 then BLG=BLG+1
if FBM=0 then FBM=1:BLG=1:FBREN=1: BITI=RIT(I3)
else
FBREN=0
end if
end if
next I3
if FBM=1 then BSU=BSU+1: BSUL(BSU,0)=0:BSUL(BSU,1)=BLG:BSUL(BSU,2)=BITI
return
::

```

```

*MXPLRIYO
' -- par. TYAB,RETUN, DR --
' MXPL_motome
MXPL=0:MXKAT=0: FCN=0:KAT=0:CNL=0
for IK=1 to RETUN
  if (KPLT(IK)>=2)and(FCN=1) then CNL=CNL+1
  if (KPLT(IK)>=2)and(FCN=0) then FCN=1:KAT=IK:CNL=1
  if (KPLT(IK)<=1)and(FCN=1) then
    if CNL>MXPL then MXPL=CNL:MXKAT=KAT
    FCN=0:CNL=0
  end if
next IK
if (FCN=1)and(CNL>MXPL) then MXPL=CNL:MXKAT=KAT
' jyoken_sirabe
if MXPL>=2 then
  MNKPLT=1000: MXSTKR=0
  for I3=MXKAT to MXKAT+MXPL-1
    if KPLT(I3)<MNKPLT then MNKPLT=KPLT(I3)
    if (TYAB=2)and(STKR(I3)>MXSTKR) then MXSTKR=STKR(I3)
  next I3
  ' hantei
  if TYAB=1 then I3ST=1:I3LT=MNKPLT-1 else I3ST=MXSTKR:I3LT=MNKPLT
  for I3=I3ST to I3LT
    TY=KTY:G=KG+I3*DR
    if KASE(TY,G)=0 then
      CHTY=4: MCS=1:MPOTI=RIT(MXKAT):POTI=MPOTI: gosub *PIECEJYOHO
      if KANOSU=0 then gosub *OVTDMSTI: return
    end if
  next I3
end if
return
::
*OVTDMSTI
' -- par. TYAB,OVTDSYOTY --
gosub *SPJYUNBI
if OVTDSYOTY=1 then
  if TYAB=1 then MSTI=-1:UMEIT1=1:UMEIT2=SL:gosub *KUROSIROMASUUME
  if TYAB=2 then PN=1:PRL(1)=MKPL: gosub *PIECEIRE
  else ' OVTDSYOTY=2
    if (I1=1)and(TYAB=1) then SRIT=APL+1
    if ((I1=1)and(TYAB=2))or((I1=2)and(TYAB=1)) then SRIT=RETUN
    SP(SRIT)=1: HSU=HSU+1:HLT(HSU)=SRIT
  end if
  gosub *EMAKAKIKAE: FTYHUKUGO=1
  return
::
*OVERTOBIDASICHB
' -- par. RETUN,OVTDSYOTY,RIT( ) --

```

```

TYAB=2
for I2=-1 to 1 step 2
DR=I2: ' STKR,KPLT_sakusei
for I3=1 to RETUN
TY=TZ:G=RIT(I3): POTI=KG+DR
if TY=0 then LTI=EMA(POTI,G) else LTI=EMA(G,POTI)
KPLT(I3)=0
if LTI<>-1 then
if TY=0 then EMA(KG,G)=-1 else EMA(G,KG)=-1 ' kari_oki
if I2=-1 then MCS=2 else MCS=1
CHTY=1: MBMLG=1:gosub *PIECEJYOHO
if TY=0 then EMA(KG,G)=0 else EMA(G,KG)=0 ' modosi
if MNKNLG>=2 then
FGO=0
for IK=1 to MNKNLG
if TY=0 then LTI1=EMA(KG+IK*DR,G) else LTI1=EMA(G,KG+IK*DR)
if LTI1=1 then FGO=1: exit for
next IK
if FGO=1 then KPLT(I3)=MNKNLG:STKR(I3)=IK
end if
end if
next I3
' MXPL_riyu
gosub *MXPLRIYO
if FTYHUKUGO=1 then return
next I2
return
::
*KUBUNTIJIME
RIT(1)=SITI+S1-1:RIT(2)=SITI+KPL
for I2=-1 to 1 step 2
FTDSU=0: DR=-I2 ' tobidasi_sirabe
for I1=1 to 2
if KTY=0 then LTI=EMA(RIT(I1),KG+I2) else LTI=EMA(KG+I2,RIT(I1))
if LTI=-1 then
if I2=-1 then MCS=1 else MCS=2
CHTY=1:TY=TZ:G=RIT(I1):POTI=KG:MBMLG=1:gosub *PIECEJYOHO
if MNKNLG>=2 then
for I3=1 to MNKNLG-1
if KTY=0 then LTI1=EMA(RIT(I1),KG+I3*DR) else LTI1=EMA(KG+I3*DR,RIT(I1))
if LTI1<>0 then exit for
next I3
if I3-1>=1 then FTDSU=FTDSU+1:KPLT(I1)=I3-1
end if
end if
next I1
'
if FTDSU=2 then

```



```

MINPL=KPLT(1): if KPLT(2)<MINPL then MINPL=KPLT(2)
for I3=1 to MINPL
TY=KTY:G=KG+I3*DR
gosub *ETISAKUSEI:gosub *WAKUKIME ' STPAN,LTPAN_ketu
' retu-kansei_sirabe
FRKSU=0
for CHN=1 to 2
ETI(RIT(CHN))=1 ' karioki
WAKUTY=1:gosub *WAKUKIME: WAKUTY=0
ETI(RIT(CHN))=0 ' modosi
gosub *RTKANHAN
if FRTKAN=1 then
FRKSU=FRKSU+1: KBL(CHN,0)=KBLSU: BRK(CHN,0)=BRKBN
for I4=1 to KBLSU
KBL(CHN,I4)=BMAL(I4)
if I4<=BRKBN then BRK(CHN,I4)=STBAN+I4-1 else BRK(CHN,I4)=STBAN+I4
next I4
end if
next CHN
' hantei
if FRKSU=2 then
for I9=STPAN to LTPAN
CS=1:SL=WKL(I9):SITI=GEN(I9)
for I1=1 to 2
' I9 ==> BLI9
BLI9=0
for GBI=1 to KBL(I1,0)
if I9=KBL(I1,GBI) then BLI9=GBI: exit for
next GBI
' SP1( ),SP2( )_sakusei
if BLI9=0 then for UU=0 to SL+1:SP(UU)=-1:next UU
if BLI9>0 then
if BLI9=BRK(I1,0) then IRETY=2:SBAN=I9 else IRETY=1
gosub *IRETYSP
end if
for I4=1 to SL
if I1=1 then SP1(I4)=SP(I4) else SP2(I4)=SP(I4)
next I4
next I1
' EMA_kakikae
HSU=0: gosub *SPSAKUSEI
for I4=1 to SL
if (SP(I4)=0)and(SP1(I4)=SP2(I4))and(SP1(I4)<>0) then
SP(I4)=SP1(I4): HSU=HSU+1:HLT(HSU)=I4
end if
next I4
if HSU>=1 then gosub *EMAKAKIKAE:FTYHUKUGO=1
next I9

```

```

    if FTYHUKUGO=1 then return
  end if
next I3
end if
next I2
return
::
*IRETYSP
ETI(RIT(I1))=1 ' karioki
gosub *SPSAKUSEI
ETI(RIT(I1))=0 ' modosi
BK0=BRK(I1,BLI9):PLG=EPL(TY,G,BK0):PLG1=PLG:PLG2=EPL(TY,G,BK0+1)
if IRETY=1 then gosub *PIECE1SET
if IRETY=2 then gosub *PIECE2SET
return
::
*AKIMASUSU
AKIMN=0
for J=1 TO GR(0):for I=1 TO GR(1)
  if EMA(I,J)=0 then AKIMN=AKIMN+1
next I,J
return
::
*MUARICH
for TY=0 to 1:for G=1 to GR(TY)
  gosub *ETISAKUSEI:gosub *WAKUKIME
  if KASE(TY,G)=1 then
    if WKN<>EPL(TY,G,0) then FMU=1: return
    gosub *MUARCH1: if FMU=1 then return
  else ' KASE(TY,G)=0
    ' check_1
    gosub *MUARCH1 : if FMU=1 then return
    ' check_2
    gosub *KBLOCKSU: if KBLSU>BMASUN then FMU=1: return
    SPLWA=0: BMLWA=BMASUN-NPN
    for I9=STPAN to LTPAN: SPLWA=SPLWA+WKL(I9): next I9
    for I9=STBAN to LTBAN: BMLWA=BMLWA+EPL(TY,G,I9): next I9
    if BMLWA>SPLWA then FMU=1: return
    ' check_3
    gosub *PIECEITI
    if LEVEL>=4 then gosub *PITITIJIME
    for I9=STPAN to LTPAN
      if BMA(I9)=1 then
        CS=1:SL=WKL(I9):SITI=GEN(I9):ZGEN=1: gosub *SPSAKUSEI:gosub *MASUNARABI
        for I8=1 to BNSU
          BIT1=BNIT(I8)-1:BMLG=BNLT(I8):BG1=STBAN:BG2=LTBAN:gosub *PIECEGAITOSU
          if ((TML(I9)=1)or(LEVEL>=4))and(KANOSU=0) then FMU=1: return
        next I8
      end if
    next I9
  end if
end for
end for

```

```

    end if
  next I9
end if
next G,TY
return
::
*MUARCH1
if KASE(TY,G)=0 then ARICH1=STPAN-1:ARICH2=LTPAN+1 else ARICH1=WKN:ARICH2=1
for I9=1 to ARICH1
  if EPL(TY,G,I9)<>WKL(I9) then FMU=1: return
next I9
for I9=WKN to ARICH2 step -1
  PBAN=EPL(TY,G,0)-WKN+I9: if EPL(TY,G,PBAN)<>WKL(I9) then FMU=1: return
next I9
return
::
*MUJYUNSYORI
FMSTOP=0
if FMUSY=0 then
  FMUSY=1
  gosub *STARTSYORI
  if FMSTOP=1 then return
else ' FMUSY=1
  ' hikae_modosu
  for J=1 to GR(0):for I=1 to GR(1): EMA(I,J)=HEMA(I,J): next I,J
  for I=0 to 1:for J=1 to GR(1): KASE(I,J)=HKASE(I,J): next J,I
  ' mujyun_ari?
  if FMU=1 then
    FMUSY=0
    ' atai_set
    LVKSU(KHL(KVA,7))=LVKSU(KHL(KVA,7))+1
    EMA(XMUT,YMUT)=-AMUT: CH(0,YMUT)=1:CH(1,XMUT)=1
    ' keizoku_jyunbi
    if KVA=1 then return
    if FMUREN=0 then FMUREN=1: MUDR=KHL(KVA,0)
    XMUT=XMUT+ZY(MUDR):YMUT=YMUT+ZT(MUDR)
  else ' FMU=0
    if FMUREN=1 then
      FMUREN=0: gosub *STARTSYORI
      if FMSTOP=1 then return
    else
      KVA=KVA+1
      if KVA>KHN then FMSTOP=1 else gosub *ATAISET
    end if
  end if
end if
return
::

```

```

*STARTSYORI
' hikae_toru
for J=1 to GR(0):for I=1 to GR(1): HEMA(I,J)=EMA(I,J): next I,J
for I=0 to 1:for J=1 to GR(I): HKASE(I,J)=KASE(I,J): next J,I
' atai_set
if (FMUREN=0)or(EMA(XMUT,YMUT)<>0) then
  FMUREN=0
  gosub *KOHOLIST ' koho-list_sakusei
  if KHN=0 then FMSTOP=1: return
  KVA=1
end if
gosub *ATAISET
return
::
*ATAISET
' -- par. FMUREN, (KVA),(XMUT,YMUT,AT) --
if FMUREN=0 then
  TY=KHL(KVA,1):G=KHL(KVA,2):XYITI=KHL(KVA,3):AMUT=KHL(KVA,5)
  if TY=0 then XMUT=XYITI:YMUT=G else XMUT=G:YMUT=XYITI
end if
' set
EMA(XMUT,YMUT)=AMUT: CH(0,YMUT)=1:CH(1,XMUT)=1
return
::
*KOHOLIST
KHN=0
LVK=3: ' level3_list
KHLIC=3: KHLV3TY=1:gosub *KHL34LIST
gosub *KHLSORT
if LEVEL=3 then return
LVK=4: ' level4_list
KHLIC=4: gosub *KHL34LIST
if LEVEL=4 then return
LVK=5: ' level5_list
for TY=0 to 1
  gosub *UDLRDAN: DANSU=int((DRDAN-ULDAN)/2)
  for DANI=1 to DANSU:for G2=1 to 2
    if G2=1 then G=ULDAN+DANI else G=DRDAN-DANI
    if KASE(TY,G)=0 then
      gosub *ETISAKUSEI:gosub *WAKUKIME
      gosub *KHSKUSEIA ' case_A
      gosub *KHSKUSEIB ' case_B
    end if
  next G2,DANI
next TY
return
::
*KHL34LIST

```

```

' -- par. KHLIC, (KHLV3TY) --
for TY=0 to 1
  gosub *UDLRDAN
  for G2=1 to 2
    if G2=1 then G=ULDAN else G=DRDAN
    gosub *ETISAKUSEI;gosub *WAKUKIME
    select case KHLIC
      case 3
        for CS=1 to 2
          if CS=1 then SITI=START:GBAN=STBAN else SITI=LAST:GBAN=LTBAN
          if KHLV3TY=1 then
            XYC=SITI:AT=1:KH0=fnDR(CS):KH4=EPL(TY,G,GBAN):gosub *KHTISET
            if (FDBL=1)and(KH4>KHL(DBAN,4)) then KHL(DBAN,4)=KH4
            else ' KHLV3TY=2
              if G2=1 then DR=1 else DR=-1
              KHN=KHN+1: KHL(KHN,1)=TY:KHL(KHN,2)=G:KHL(KHN,3)=SITI:
              KHL(KHN,4)=EPL(TY,G,GBAN):KHL(KHN,5)=CS:KHL(KHN,6)=DR
            end if
          next CS
        case 4
          if NPN>=2 then gosub *KHSKUSEIA ' case_A
          gosub *KHSKUSEIB ' case_B
        end select
      next G2,TY
    return
  ::
*KHTISET
' -- par. XYC,AT, KH0,KH4 --
gosub *KHLDBCH
if FDBL=0 then
  KHN=KHN+1: KHL(KHN,0)=KH0:KHL(KHN,1)=TY:KHL(KHN,2)=G
  KHL(KHN,3)=XYC:KHL(KHN,4)=KH4:KHL(KHN,5)=AT:KHL(KHN,7)=LVK
end if
return
::
*KHLDBCH
' -- par. XYC,AT --
FDBL=0
for IK=1 to KHN
  if TY=0 then XT1=XYC:YT1=G else XT1=G:YT1=XYC
  if KHL(IK,1)=0 then XT2=KHL(IK,3):YT2=KHL(IK,2) else XT2=KHL(IK,2):YT2=KHL(IK,3)
  if (XT1=XT2)and(YT1=YT2)and(AT=KHL(IK,5)) then FDBL=1:DBAN=IK: return
next IK
return
::
*KHSKUSEIA
for I9=STPAN to LTPAN
  if TML(I9)=0 then

```

```

for CS=1 to 2
  if CS=1 then SITI=GEN(I9):ZU=1 else SITI=GEN(I9)+WKL(I9)-1:ZU=-1
  ' XYITI_kime
  REPIT=SITI:gosub *XYITI
  XYC=REPIT:AT=1:KH0=fnDR(CS):KH4=0:gosub *KHTISET
  next CS
end if
next I9
return
::
*KHSAKUSEIB
for CS=1 to 2
  if CS=1 then SITI=START:GBAN=STBAN:ZGEN=1 else SITI=LAST:GBAN=LTBAN:ZGEN=-1
  ' XYITI_kime
  REPIT=SITI+(EPL(TY,G,GBAN)-1)*ZGEN:ZU=-ZGEN:gosub *XYITI
  XYC=REPIT:AT=-1:KH0=RD(fnDR(CS)):KH4=0:gosub *KHTISET
  next CS
  return
  ::
*XYITI
' -- par. REPIT,ZU --
if TY=0 then ETI=EMA(REPIT,G) else ETI=EMA(G,REPIT)
while ETI=1
  REPIT=REPIT+ZU: if TY=0 then ETI=EMA(REPIT,G) else ETI=EMA(G,REPIT)
wend
return
::
*KHLSORT
for IJ=1 to KHN-1:for IK=IJ+1 to KHN
  if KHL(IJ,4)<KHL(IK,4) then
    for PQ=0 to 7: swap KHL(IJ,PQ),KHL(IK,PQ): next PQ
  end if
next IK,IJ
return

```

## 5. 問題 [2] ～ [4] の結果とその考察

サイズの大きさにより、3種類の問題集を取上げた。問題集 [2] は小型サイズ（マス数 400～875）、問題集 [3] は中型サイズ（マス数 2000～4125）、問題集 [4] は大型サイズ（マス数 9000～14000）のもののみ使用した。また、これら3種類では解の難易を示す尺度（表示難易として利用）として、問題集 [2] では難易度 1～3、問題集 [3] では難易度 2～5、問題集 [4] では難易度 4～10（一部不定のものがある）がつけられている。

さて、3.1 で示したように、解の難易を決める基準として、

- ① レベル 1～5 の 5 段階とする、
- ② 問題が本来持っている難しさ（論理思考の高い低い）に加えて、解くときの手間とか面

倒くさを考慮する

であった。すでに問題本来のもつ難しさについては「手法」によって決定しているので、解くときの手間の基準化とそれを加えた新しい難易度基準を作成することである。

### 5.1 手法だけによる解結果

3.2で示した「具体的手法のリストとそのレベル」により問題を解いたときの、各サイズのレベル結果[表中の数字は題数を表す]はつぎのようにまとめられる。

#### 1) 小型サイズ (200 題)

表示 難易	解 レ ベ ル					計
	[1]	[2]	[3]	[4]	[5]	
1	3	27	8			38
2	3	44	48	2	1	98
3	1	9	44	5	5	64

#### 2) 中型サイズ (287 題)

表示 難易	解 レ ベ ル					計
	[1]	[2]	[3]	[4]	[5]	
2		4	4			8
3	1	13	27	12		53
4		24	104	43	2	173
5		7	30	12	4	53

#### 3) 大型サイズ (32 題)

解 レベル	表 示 難 易								計
	4	5	6	7	8	9	10	不定	
3		6	2	2				1	11
4	1	3	2	7	2	1	1	2	19
5					2				2
計	1	9	4	9	4	1	1	3	32

(注意) 例えば、小型サイズで「難易1(★)」と表示されている38題は、3題がレベル1で解かれ、30題がレベル2、5題がレベル3で解ける。また、大型サイズは表示難易と解レベルの示し方が逆になっている(不定とは、難易度が隠されている問題)。

この結果からつぎのことがいえる。

- (1) 小型サイズでは、解レベルは表示難易より高くなっている。

これは、3段階で付けられたものが5段階レベルに対応されているので妥当なのかもしれない。例えば、表示難易1に解レベル（1と2の1/3）を、表示難易2に解レベル（2の2/3と3の半分）を、表示難易3に解レベル（3の半分と4と5）を対応づけるなら、表示難易と解レベルの適合率は表示難易1で31.6%、表示難易2で54.4%、表示難易3で50%となる。

(2) 中型と大型サイズでは、解レベルは表示難易より低くなっている。特に、難易度の高いケースでそれが相当目立つ。例えば、中型サイズでは、レベルが難易度より低い比率は、難易度4で71.1%、難易度5で94.3%である。

従って、中型サイズ以上で、解くときの手間を加えたレベルアップが必要である。

## 5.2 解法の手間を反映するレベルアップ基準

解法の手間を反映する尺度として、解レベルを決定付ける手法処理を何回行なったかをとる。ただし、基本処理の「ピース入れ」処理だけはそのカウントから除外するのが妥当と判断した。

この数がある限度値を超えたとき、解レベルを1だけアップさせるとし、次の原則をおく。

〔原則〕①解レベルのそれぞれに応じて、カウント数に基づく度数分布表を作成する。そして、レベルアップの限度値はある区分の開始値とする（切りのよい値となる）。

② 限度値の満たすべき性質として次をとる。

(i) 表示難易度 $\leq$ 解レベルの場合、解レベルの保持題数 $\geq$ 解レベルのアップ題数、

(ii) 表示難易度 $>$ 解レベルの場合、解レベルの保持題数 $\leq$ 解レベルのアップ題数、

さて、限度値を決めるため、問題を解いて得られた結果をつぎに示す。

1) 中型サイズ (281 題)

○ 解レベル：2

区分	度数	表示難易			
		2	3	4	5
0 ~ 49	2	-	-	-	-
50 ~ 99	11	1/3	0/13	1/23	0/7
100 ~ 149	21	2/2	5/8	6/18	0/7
150 ~	14	3/1	11/2	14/10	6/1
計	48				



○ 解レベル：3

区分	度数	表示難易			
		2	3	4	5
0 ~ 24	7	-	-	-	-
25 ~ 49	28	1/3	3/24	3/101	0/30
50 ~ 74	40	2/2	11/16	20/84	2/28
75 ~ 99	40	3/1	17/10	46/58	9/21
100 ~ 124	27	3/1	23/4	71/33	18/12
125 ~	23	4/0	24/3	89/15	25/5
計	165				

○ 解レベル：4

区分	度数	表示難易		
		3	4	5
0 ~ 9	11	-	-	-
10 ~ 19	25	5/7	4/39	2/10
20 ~ 29	14	10/2	20/23	6/6
30 ~ 39	6	10/2	30/13	10/2
40 ~ 49	3	11/1	35/8	10/2
50 ~ 59	4	11/1	38/5	10/2
60 ~	4	11/1	41/2	11/1
計	67			

(注意) この表で、例えば、解レベル3の区分50～74について、限度値は50となり、その際、表示難易3の保持題数とアップ題数はそれぞれ11題、16題となる。

2) 大型サイズ (30題)

○ 解レベル：3

区分	度数
100 ~ 199	1
200 ~ 299	2
300 ~ 399	1
400 ~	4
計	8

○ 解レベル：4

区分	度数	表示難易
100 ~ 149	3	0/22
150 ~ 199	3	0/22
200 ~ 249	6	0/22
250 ~ 299	6	1/21
300 ~	4	1/21
計	22	

この結果と原則に基づき次の基準が決まる。

(1) 解レベル2の限度値は100となる。

この時に原則が成り立つだけでなく、その他の場合は原則が成り立たないから。例えば、限度値が50なら、表示難易2において、解レベルの保持題数は1で解レベルのアップ題数3より小さく原則②(i)が成り立たない。

(2) 解レベル3の限度値は75となる。

この時に原則が成り立つだけでなく、その他の場合は原則が成り立たないから。例えば、限度値が50なら、表示難易3において、解レベルの保持題数は11で解レベルのアップ題数16より小さく原則②(i)が成り立たない。

また、大型サイズに適用すればアップ率は100%となり、この場合にもマッチする。

(3) 解レベル4の限度値は25がふさわしい。

原則が成り立つ条件は、①表示難易3では限度値が20以上、②表示難易4では、20以下ならダメだが、30以上ならよい、③表示難易5では20以上ならよい。この3つのことから、限度値として切りのよい値25が適当と考えられる。

また、大型サイズでは、原則から200以下が考えられ、中型サイズで決まった基準はこれにマッチする。

[レベルアップの基準]

(1) 解レベルが1の時、マス数が2,000以上ならレベルを1上げる。

(2) 解レベルが2の時、解決定の処理回数が100回以上ならレベルを1上げる。

(3) 解レベルが3の時、解決定の処理回数が50回以上ならレベルを1上げる。

(4) 解レベルが4の時、解決定の処理回数が25回以上ならレベルを1上げる。

### 5.3 新しい基準によるレベル付けと表示難易との適合性

新しい基準によるレベル付けでの問題の解結果[題数]はつぎのようにまとめられる。

1) 小型サイズ (200題)

表示 難易	解 レ ベ ル					計
	[1]	[2]	[3]	[4]	[5]	
1	3	27	8			38
2	3	44	48	2	1	98
3	1	9	42	7	5	64

## 2) 中型サイズ (287 題)

表示 難易	解 レ ベ ル				計
	[ 2 ]	[ 3 ]	[ 4 ]	[ 5 ]	
2	2	4	2		8
3	6	19	26	2	53
4	6	38	110	19	173
5		9	35	9	53
計	14	70	173	30	287

## 3) 大型サイズ (29 題)

解 レベル	表 示 難 易						計
	4	5	6	7	8	9,10	
4		6	2	2			10
5	1	3	2	7	4	2	19
計	1	9	4	9	4	2	29

この結果からつぎのことがいえる。

(1) 小型サイズの結果。表示難易2までは新基準になっても変化がなく、表示難易3で、解レベル3の題数が2減り、解レベル4の題数が2増えている。

従って、新しい基準による本質的な変化はないと見てよい。

(2) 中型サイズの結果。解レベルが全体的にアップした。その結果として、表示難易2,3では解レベルが表示難易より高くなる度合いがさらに強まり、難易4では一致度が63.6%と相当高まり、また難易5では表示難易が2以上離れる率が17%と相当低くなった。

なお、結果を全体的に見れば、表示難易度が低い場合は解レベルの方が表示難易より大きく、表示難易度が高い場合は解レベルの方が表示難易より小さくなる。

(3) 大型サイズの結果。全体的に表示難易と解レベルの一致度が高まった。また、表示難易が6まで（データ数1の難易度4は除けて）において解レベルが低くなる傾向が見られる。この結果は、10段階で付けられたものが5段階レベルに対応されているので妥当ともいえる。

#### 5.4 表示難易と解レベルの対応に関する考察

5.3の結果はあまり好ましいものではない。しかし、これは解レベルの改良によって解決できるというより、根本的に構造的なものと思われる。以下、この点について考察したい。まず、5.3の結果の特徴をあげる。

(1) 表示難易度が低い場合は解レベルの方が表示難易より大きく、逆に表示難易度が高い場合は解レベルの方が表示難易より小さくなる。

(2) サイズが小さい場合は解レベルの方が表示難易より大きく、逆にサイズが大きい場合は解レベルの方が表示難易より小さくなる。なお、中間サイズでは、特別にやさしいものや難しいものを除いて、ほとんどの問題がやや難しいと判断される傾向がある。

以上の2つの特徴を説明するため、つぎの仮説をとる。

【仮説】 マスの数が多くなると、

・人間の側から見ると、適用できる手法を探す手間が増えること（コンピュータによる手法処理ではこれはない）により、表示難易は高くなる。

・コンピュータの解処理から見ると、適用できる手法が増えるので、解レベルは低くなる。すると、この仮説からつぎのように考えられる。

① サイズに関する (2) について。サイズが大きい場合は上の仮説から、またサイズが小さい場合は、手法への関心が強まることからその難しさが薄まり、一方で手間の方は無視できるとみなせるから。

② 中間サイズでは上の仮説がかなり効果を発揮すると考えられる。適用できる手法が増えることから解レベルは下がり、同時に人間の手間の増加により表示難易はあがる。従って、多くの問題がやや難しいと判断されるのである。

## 文 献

- [1] 佐藤金吾, 「イラストパズルを解くコンピュータプログラム」, 法政大学多摩研究報告, 15 (2000)
- [2] パズル・ポシェット 『イラストロジック③,④』, 日本文芸社, 1996～1999年
- [3] 『ジャンボロジック⑤,⑥,⑧』, 日本文芸社, 1999～2000年
- [4] 『超特大ロジックパラダイス』, 学習研究社, 1999年
  - 『イラストロジック スーパージャンボ③』, 学習研究社, 1998年
  - 『超難問ロジック①～④』, 学習研究社, 1999～2002年
  - 『イラストロジック傑作集 1,2』, コスミックスインターナショナル, 2001年
  - 『ロジックプラザ DX1,2』, コスミックスインターナショナル, 2001年

他