学習指導要領（3）- 知・技 - イ
学習指導要領（3）- 思・判・表 - イ
学習内容（3）- イ アルゴリズムとプログラム

情報 III

順列とは、有限個の事象の重複のない並び順である。多くの実社会の問題は、最大価値を与える順列や、与えられた制約を満足する順列を求める問題に帰結できる。膨大な候補から正解や最適解を求めることや、そもそも正解が存在するかを人手で判断するのは、時間が掛かる上に間違えることもあるので、コンピュータで処理することが有効である。

(ア) 次の文章の空欄 (39)(40)(41)(42)(43)(44) ～ (57)(58)(59)(60)(61)(62) には、あてはまるもっとも適切な数字を解答欄にマークしなさい。

ある工業製品の作業工程が A_1, A_2, A_3, A_4, A_5 の5工程あり、作業員が W_1, W_2, W_3, W_4, W_5 の5名いて、各作業員が1つの工程を担当し、作業工程順と担当作業員によって生産効率が変わるという状況で、作業工程順と担当作業員を決定せねばならないとしよう。

作業工程の順列は (39)(40)(41)(42)(43)(44) 通りあり、作業工程順と担当作業員のあらゆる候補数は (45)(46)(47)(48)(49)(50) 通りである。

もし作業工程 A_4 は必ず A_5 の直後という制約がある場合には、その条件を満たす作業工程順と担当作業員の候補数は (51)(52)(53)(54)(55)(56) 通り、作業工程 A_4 は必ず A_5 の直後という制約に加えて、作業工程 A_3 は必ず W_2 以外の作業員が行う制約がある場合には、この2つの制約を満足する作業工程順と担当作業員の候補数は (57)(58)(59)(60)(61)(62) 通りとなる。このようにもともとの検索範囲が膨大であっても、制約条件をプログラム中でうまく利用することで、検索範囲を狭め結果的に早く計算できる。

(イ) 次の文章の空欄 (63)(64) ～ (87)(88) には、下の選択肢から最も適切なものを選び、その番号を解答欄にマークしなさい。

n 個の事象に対して順列を全て列記するプログラムの開発を考える。それぞれの事象は順列の中で一回しか現れないことを利用し、初期状態から2つの事象を入れ替える手続きを繰り返して、重複することなく順列を列記するアプローチを取る。事象の順列をプログラムで表すために、事象数と同数の変数 $H_1, \ldots, H_n$ を用いる。H は添え字1を最下位として、添え字が大きくなることを上位と表現する。この変数に事象を漏れなく、同じ順列を与えないように注意しながら代入することで順列を全て列記する。

例えば、3つの事象 A_1, A_2, A_3 がそれぞれ H_1, H_2, H_3 に代入されている順列を初期状態とした場合の、事象の入れ替えによる順列列記の例を図1に示す。図中の二重枠は前の状態から変化がない変数を表している。事象が n 個の時、同じ順列が現れないように工夫するなら、初期状態から (63)(64) 回事象の入れ替えを行い、初期状態の順列と、入れ替えを行う度にその時点の順列を出力することで、順列を全て列記できる。

$A_1, \ldots, A_{n-1}$ の $n-1$ 個の事象に対する順列を全て列記するための処理を処理 P_{n-1} とした時、事象 A_n を加えた n 個の事象の順列を全て列記する処理 P_n は、H_n に A_n を代入した状態で H_1 から H_{n-1} に対して処理 P_{n-1} を実施し、次に H_1 から H_{n-1} に代入されている事象から一つ選んで H_n の事象と入

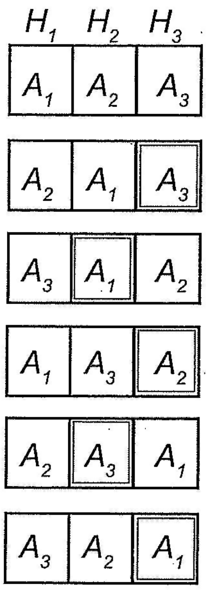

図１　順列列記の例

れ替えて処理 P_{n-1} を実施する、という手続きを繰り返すことに置き換えられる。H_n の事象入れ替えの際に、A_1 から A_n の事象が H_n に１回だけ代入されるようにすれば、同じ順列が現れることはない。つまり処理 P_n は H_n に代入される事象を入れ替えながら、処理 P_{n-1} を (65)(66) 回実施することと等価である。

処理 P_n において H_n に代入する事象の選択には複数の方法があるが、以下のアルゴリズムは計算効率が良い。

- n が奇数の場合には処理 P_{n-1} 実施後に H_n と H_1 の事象を入れ替える
- n が偶数の場合には最初の処理 P_{n-1} 実施後に H_1 と H_n の事象を入れ替え、次の処理 P_{n-1} 実施後には H_2 と H_n の事象を入れ替えというように H_1 から順に入れ替え場所を上位に移動する

事象数が n の場合、H_n の事象入れ替え操作は初期状態から (67)(68) 回行うことになる。

同様に処理 P_{n-1} を処理 P_{n-2} の繰り返しに置き換え、処理 P_{n-2} を処理 P_{n-3} の繰り返しに置き換え、というように、扱う事象数を減らした処理の繰り返しに順次置き換え、P_1 に至れば事象が１個しかないため、処理 P_0 は実施しない。

上記のアルゴリズムを適用して、$A_1, \ldots, A_n$ がそれぞれ $H_1, \ldots, H_n$ に代入されている初期状態から順列を全て列記する場合、H_n は、一定回数順列が出力される間同じ事象が留まり、それから次の事象へと順次変化する。図１はこのアルゴリズムを用いており、H_3 に配置される事象は２回連続で留まりながら、列記が進むにつれ A_3, A_2, A_1 と変遷している。

様々な n に対して、H_n に配置される事象が A_i の次に、A_i とは異なる事象 A_j となる変遷を $A_i \to A_j$ と表現するとして、上記のアルゴリズムを実施した場合に H_n に代入される事象の変遷に関する下表を完成させなさい。

事象数 (n)	H_n に代入される事象の変遷
2	$A_2 \to A_1$
3	$A_3 \to A_2 \to A_1$
4	$A_4 \to \boxed{(69)\,\vert\,(70)} \to \boxed{(71)\,\vert\,(72)} \to A_1$
$\vdots$	$\vdots$
8	$A_8 \to \boxed{(73)\,\vert\,(74)} \to \boxed{(75)\,\vert\,(76)} \to \boxed{(77)\,\vert\,(78)} \to \boxed{(79)\,\vert\,(80)} \to \boxed{(81)\,\vert\,(82)} \to \boxed{(83)\,\vert\,(84)} \to A_1$
$\vdots$	$\vdots$
11	$A_{11} \to \boxed{(85)\,\vert\,(86)} \to \boxed{(87)\,\vert\,(88)} \to \cdots$

【$\boxed{(63)\,\vert\,(64)}$〜$\boxed{(87)\,\vert\,(88)}$ の選択肢】

(11) A_1 (12) A_2 (13) A_3 (14) A_4

(15) A_5 (16) A_6 (17) A_7 (18) A_8

(19) A_9 (20) A_{10} (21) A_{11} (22) n

(23) $n-1$ (24) $n+1$ (25) $n!$ (26) $n!-1$

(27) $(n-1)!$ (28) $n-i$ (29) $n+i$