

# 順列生成の開発Ⅲ

2013.05.23

## 問題 1 ~ 6

途中省略。

### 【補足 2】部分集合の生成順序について

部分集合の生成において、生成順序を集合の要素数の昇順となるように生成したい場合がある。

例えば、 $N=3$  の場合

0 0 0 → 空集合  
1 0 0 → {1}  
0 1 0 → {2}  
0 0 1 → {3}  
1 1 0 → {1,2}  
1 0 1 → {1,3}  
0 1 1 → {2,3}  
1 1 1 → {1,2,3}

プログラムリストは別紙 8 参照。

これは、**次週のテーマ「マッチ棒問題」**で使用予定である。マッチ棒を列単位で割り当てていくとき、取り除くマッチ棒の数がそろっていると、マッチ棒の累計が上限値を超えた時点でそれ以降の探索を打ち切ることができるため、都合がよい。

### 番外 組み合わせ型完全ハッシュ関数について

組み合わせ型ハッシュ関数は、順列型ハッシュ関数と同様にパズル Solver でよく使われるハッシュ関数である。

#### 【組み合わせ型完全ハッシュ関数】

$0 \sim (n-1)$  の  $n$  個の数字から  $r$  個を取り出す組み合わせに  $0 \sim {}_n C_r - 1$  のハッシュ値を割り当てる関数を考える。

$r$  個の数字を  $0 \leq a_1 < a_2 < \dots < a_r$  とすると、ハッシュ値を次の式で定義する。

$$h = a_r C_r + a_{r-1} C_{r-1} + \dots + a_1 C_1 \quad \text{ただし、} n C_r = 0 \quad (n < r)$$

#### 【例】

$0 \sim 4$  の 5 個の数字から 2 個を選ぶ組み合わせは 10 通りある。それぞれの組み合わせに  $0 \sim 9$  の数字を割り当てる場合について説明する。

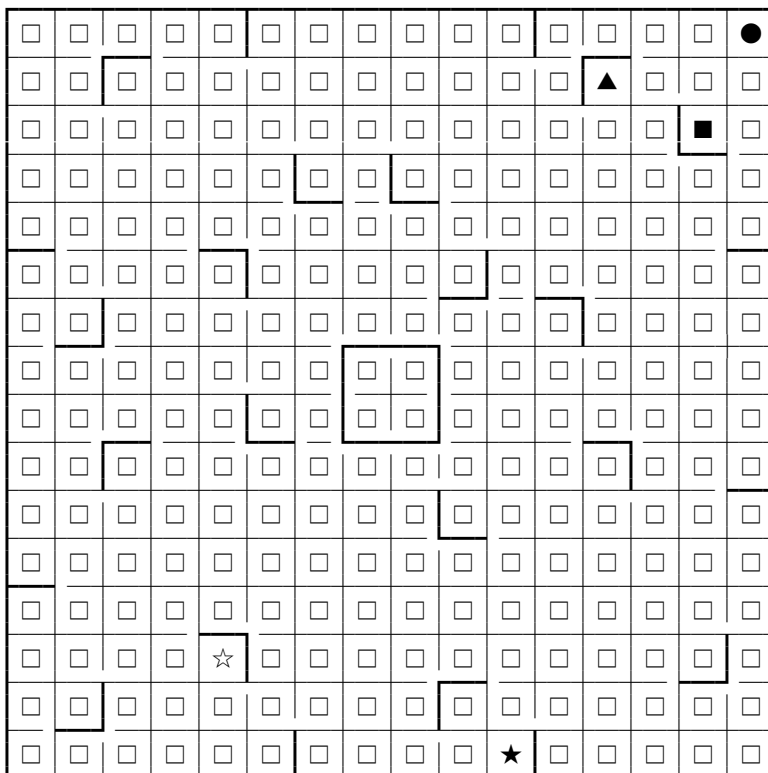
0 1 2 3 4    選択した数字    ハッシュ値の計算

1 1 0 0 0	0,1	-->	$1C_2 + 0C_1 = 0$
1 0 1 0 0	0,2	-->	$2C_2 + 0C_1 = 1$
0 1 1 0 0	1,2		
1 0 0 1 0	0,3		途中略
0 1 0 1 0	1,3	-->	$3C_2 + 1C_1 = 4$
0 0 1 1 0	2,3		
1 0 0 0 1	0,4		
0 1 0 0 1	1,4		途中略
0 0 1 0 1	2,4		
0 0 0 1 1	3,4	-->	$4C_2 + 3C_1 = 9$

【実例】

GPCC の 2 0 1 0 年の問題に、ハイパーロボットの手数探索問題が提起されている。この問題は、 $16 \times 16$  の盤上に 4 台のロボットが配置される。4 台のロボットのうち 3 台のロボットの機能は同じため、区別する必要がない。従って、局面（ロボットの位置）を記述するためのハッシュ関数は組み合わせ型となる。局面の数は、 $256 * 256 C_3 = \text{約 } 707M \text{ 個}$  となり、1 Byte / 1 個とすると、32 ビットマシンでは結構厳しい容量となる。

ハイパーロボットの問題図



- ☆ : ゴールの位置
- ★ : ゴールを目指すロボット
- , ▲, ■ : 一般のロボット

また、このハッシュ関数の逆関数も性能的には気になるが、 $256\mathbf{C}_3$  の部分は逆変換の論理を組むよりもテーブルを使って求めた方が簡単である。 $(256\mathbf{C}_3=約 2.8M 個なので二次キャッシュにも乗りそう)$

以上