章末問題（2）のアルゴリズムにおいて，**1-2-2-2°** の操作

**1-2-2-2°** *Maze*[*ii,jj*]　:=　*Maze*[*i,j*] + 1　;

を下記のように変更し，各セルに番号 1，2，3 のいずれかをこの順に付けて行くようにする．

**1-2-2-2°** *Maze*[*ii,jj*]　:=　( *Maze*[*i,j*]　**mod** 3 ) + 1　;

ここで，( *k*　**mod** 3 ) は，*k* を 3 で割った余りを取る演算であり，*Maze*[*i,j*] = 3 の場合，*Maze*[*ii,jj*] には 1 が入ることになる．

図 2.11 の迷路に，スタック *S* を用いて 1，2，3 の番号を付けると，下図のようになる．

| *i* \ *j* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 0 | 0 | 0 | | | 0 | 0 | 0 |
| 1 | 2 | | | | 0 | 0 | | 0 | | |
| 2 | 3 | | 0 | | | 0 | | 0 | 0 | 0 |
| 3 | 1 | | 0 | 0 | | | | | 2 | |
| 4 | 2 | | | 0 | | 1 | 2 | 3 | 1 | |
| 5 | 3 | 1 | 2 | 3 | | 3 | | | 2 | 3 |
| 6 | | | 3 | | | 2 | 1 | | | 1 |
| 7 | 1 | | 1 | 2 | | 1 | 3 | | 3 | 2 |
| 8 | 3 | | 2 | | | | 2 | | 1 | |
| 9 | 2 | 1 | 3 | 1 | 2 | 3 | 1 | | 2 | 3 |

この図からも分かるように，セル [*i,j*] に付与された番号 *Maze*[*i,j*] が 1 であれば，[*i,j*] の隣接セルには，2 あるいは 3 の番号が付与されており，セル [0,0] の方に戻るためには，3 の番号が付与されたセルの方に行けばよい．同様に，*Maze*[*i,j*] が 2（3）であれば，[*i,j*] の隣接セルには，1 あるいは 3（1 あるいは 2）の番号が付与されており，セル [0,0] の方に戻るためには，1（2）の番号が付与されたセルの方に行けばよい．したがって，章末問題 (4) の解答例に示した手続き *Backtrack* をわずかに変更するだけで，セル [0,0] からセル [*n,m*] への道を求めることができる．

このように各セル $[i,j]$ に対して覚えておかねばならない数が高々 3 でよくなれば，$Maze[i,j]$ に要するビット数が少なくよくなる．すなわち，0,1,2,3 だけを覚えるのであれば，2 ビットでよい．それゆえ，メモリ領域の節約になる．

アルゴリズムの変更は各自試みるとよい．