

MATLAB で覆面算を解こう！！

ここでは、MATLAB によるプログラミング学習の一環として、覆面算を扱っていく。

覆面算における一般的なルール

1 つの英文字は、0～9 のいずれかで、文字が違えば対応する数字も異なる。

このようなルールを用いる覆面算で有名な問題に、

SEND+MORE=MONEY

がある。これを MATLAB で解くにはどうしたらよいであろうか？

もっとも単純なのは、総当たりで数字をチェックし、一致するか否かを調べればよい。

では、MATLAB を使いスクリプトを書いてみよう。

SEND+MORE=MONEY では、変数は、S,E,N,D,M,O,R,Y の 8 文字分で探せばよい。for 文を使い全通りを出力するスクリプトは以下になる。数字の重複チェックは、if-continue 文により行う。これにより重複なしにすべての通りの数字を出力することができる。

```
for S=1:9;for E=0:9;for N=0:9;for D=0:9
for M=1:9;for O=0:9;for R=0:9;for Y=0:9
if(S==E) continue;end
if(S==N) continue;end
if(S==D) continue;end
if(S==M) continue;end
if(S==O) continue;end
if(S==R) continue;end
if(S==Y) continue;end

if(E==N) continue;end
if(E==D) continue;end
if(E==M) continue;end
if(E==O) continue;end
if(E==R) continue;end
if(E==Y) continue;end

if(N==D) continue;end
if(N==M) continue;end
if(N==O) continue;end
if(N==R) continue;end
if(N==Y) continue;end

if(D==M) continue;end
if(D==O) continue;end
if(D==R) continue;end
if(D==Y) continue;end

if(M==O) continue;end
if(M==R) continue;end
if(M==Y) continue;end
```

```
if(O==R) continue;end
if(O==Y) continue;end

if(R==Y) continue;end
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n',S,E,N,D,M,O,R,Y);
end;end;end;end;
end;end;end;end;
```

ちなみに、S,M は、桁の最初にあるため 0 にはなりえない点に注意すること。

覆面算の計算チェックには、次の if 文を使う。

```
if 1000*S+100*E+10*N+D+1000*M+100*O+10*R+E==10000*M+1000*O+100*N+10*E+Y
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n',S,E,N,D,M,O,R,Y);
end
```

このとき、if 文の等号は、=ではなく、==を用いる点に注意する必要がある。以上をまとめると、

```
tic
for S=1:9;for E=0:9;for N=0:9;for D=0:9
for M=1:9;for O=0:9;for R=0:9;for Y=0:9
if(S==E) continue;end
if(S==N) continue;end
if(S==D) continue;end
if(S==M) continue;end
if(S==O) continue;end
if(S==R) continue;end
if(S==Y) continue;end

if(E==N) continue;end
if(E==D) continue;end
if(E==M) continue;end
if(E==O) continue;end
if(E==R) continue;end
if(E==Y) continue;end

if(N==D) continue;end
if(N==M) continue;end
if(N==O) continue;end
if(N==R) continue;end
if(N==Y) continue;end

if(D==M) continue;end
if(D==O) continue;end
if(D==R) continue;end
if(D==Y) continue;end

if(M==O) continue;end
if(M==R) continue;end
if(M==Y) continue;end
```

```
if(O==R) continue;end
if(O==Y) continue;end

if(R==Y) continue;end
if 1000*S+100*E+10*N+D+1000*M+100*O+10*R+E==10000*M+1000*O+100*N+10*E+Y
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d¥n',S,E,N,D,M,O,R,Y);
end
end;end;end;end;
end;end;end;end;

toc
```

Core2Duo では、

```
S=9;E=5;N=6;D=7;M=1;O=0;R=8;Y=2
```

経過時間は 362.013772 秒です

が実行速度であった。ちなみに、MATLAB では、コピーペーストして実行した場合と、スクリプトファイル化して実行した場合では、実行速度が変わる。当然のことながら、スクリプト化して実行した方が速くなる。上の数字は、コピーペーストして実行した例であるが、スクリプトファイル化して実行した場合には、

```
>> Untitled
```

```
S=9;E=5;N=6;D=7;M=1;O=0;R=8;Y=2
```

経過時間は 97.703874 秒です

とこれだけでも 3 倍速くなる。

また、if-continue 文の場所を以下のように変更し、スクリプトファイルとして実行すると、

```
tic
for S=1:9;for E=0:9;
if(S==E) continue;end
for N=0:9;
if(S==N) continue;end
if(E==N) continue;end
for D=0:9
if(S==D) continue;end
if(E==D) continue;end
if(N==D) continue;end
for M=1:9;
if(S==M) continue;end
if(E==M) continue;end
if(N==M) continue;end
if(D==M) continue;end
for O=0:9;
if(S==O) continue;end
if(E==O) continue;end
if(N==O) continue;end
if(D==O) continue;end
if(M==O) continue;end
for R=0:9;
if(S==R) continue;end
if(E==R) continue;end
```

```
if(N==R) continue;end
if(D==R) continue;end
if(M==R) continue;end
if(O==R) continue;end
for Y=0:9
if(S==Y) continue;end
if(E==Y) continue;end
if(N==Y) continue;end
if(D==Y) continue;end
if(M==Y) continue;end
if(O==Y) continue;end
if(R==Y) continue;end
if 1000*S+100*E+10*N+D+1000*M+100*O+10*R+E==10000*M+1000*O+100*N+10*E+Y
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n',S,E,N,D,M,O,R,Y);
end
end;end;end;end;
end;end;end;end;
toc
```

実行結果は、

>> Untitled

S=9;E=5;N=6;D=7;M=1;O=0;R=8;Y=2

経過時間は 7.549093 秒です

97.7/7.5=13 と if 文の場所を変えるだけで約 13 倍速く計算できる。速くなる理由は、なるべく for ループの外に if 文を置くことで、if でチェックする回数を減らしているからである。多重ループのアルゴリズムにおいては、かなりの効果が見込めるテクニックである。ちなみにこのテクニックは、C 言語や Java 言語でも使える。

以上より、答えは、

9567+1085=10652

として計算できる。

さらなるチューニング

MATLAB に適したスクリプトの書き方をするにはどのようにしたらよいであろうか？ if 文の中を MATLAB らしい書き方に変更した例を示す。この場合、配列演算子である、配列乗算.*と、配列べき乗演算子.^を使って計算している。計算結果は、sum 関数により合計している。

```
if sum([S,E,N,D]+[M,O,R,E]).*10.^(3:-1:0))==sum([M,O,N,E,Y]).*10.^(4:-1:0))
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n',S,E,N,D,M,O,R,Y);
end
```

このようにすると若干見栄えが良くなる。ただしこの場合は、処理速度は遅くなる。

もし、処理速度を考慮した場合には、加算を先に行い、乗算する回数を減らすとよい。

```
if (S+M-O)*1000+(E+O-N)*100+(N+R-E)*10+D+E==10000*M+Y
fprintf(' S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n' , S, E, N, D, M, O, R, Y) ;
```

end

大したことはないが・・・

また、+演算子を使って if 文の数を減らすこともできる。

```
tic
for S=1:9;for E=0:9;
if (S==E) continue;end
for N=0:9;
if (S==N)+(E==N) continue;end
for D=0:9;
if (S==D)+(E==D)+(N==D) continue;end
for M=1:9;
if (S==M)+(E==M)+(N==M)+(D==M) continue;end
for O=0:9;
if (S==O)+(E==O)+(N==O)+(D==O)+(M==O) continue;end
for R=0:9;
if (S==R)+(E==R)+(N==R)+(D==R)+(M==R)+(O==R) continue;end
for Y=0:9;
if (S==Y)+(E==Y)+(N==Y)+(D==Y)+(M==Y)+(O==Y)+(R==Y) continue;end
if (S+M-O)*1000+(E+O-N)*100+(N+R-E)*10+D+E==10000*M+Y
fprintf('S=%d;E=%d;N=%d;D=%d;M=%d;O=%d;R=%d;Y=%d\n',S,E,N,D,M,O,R,Y);
end
end;end;end;end;
end;end;end;end;
```

実行結果は、

>> Untitled

S=9;E=5;N=6;D=7;M=1;O=0;R=8;Y=2

経過時間は 4.799360 秒です

と若干改善することができる。

ちなみに、SEND+MORE=MONEY より、M は 1 であろうとある程度予想ができれば、さらに高速化ができる。この場合には、for M=1:9 のところを for M=1:1 とすれば、1 に確定することができる。ちなみに、この変更をして実行すると、

>> Untitled

S=9;E=5;N=6;D=7;M=1;O=0;R=8;Y=2

経過時間は 0.602874 秒です

として計算することができる。

補足

覆面算の例で簡単なものを例に別の解法について考えていく。なお、この方法は、原理的には、速いが、メモリを多く消費するため、SEND+MORE=MONEY の例題でなく簡単な例で解説していく。

```
ABC
- BA
-----
BB
```

という問題を考える。この問題の解き方は、

```
tic
for A=1:9
    for B=1:9
        if(A == B) continue;end
        for C=0:9
            if(A == C) continue;end
            if(B == C) continue;end
            ABC=A*100+B*10+C;
            BA=B*10+A;
            BB=B*10+B;
            if(ABC-BA == BB) [A,B,C],end
        end
    end
end
end
toc
```

で解くことができるが、もっと他の解き方はないであろうか？

MATLAB 流の解き方では、なるべく for 文を使わないほうが処理が速くなる。そこでどのように解くのか？

meshgrid 関数、ndgrid 関数を使う方法

meshgrid 関数は、例えば、2次元平面の座標を表す行列を作るときに使う。

```
>> [x,y]=meshgrid(1:3,1:3)
x =
     1     2     3
     1     2     3
     1     2     3

y =
```

1	1	1
2	2	2
3	3	3

これを使うと、 $z=x^2+y^2$ などの 3 次元グラフが簡単に作れる。

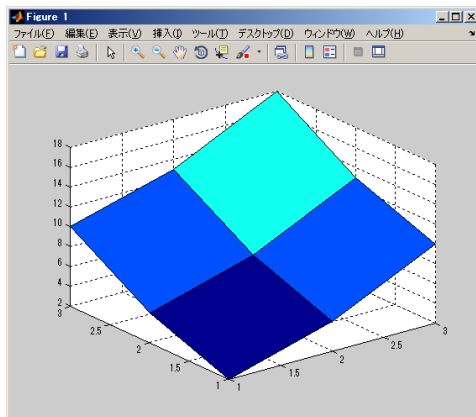
```
>> z=x.^2+y.^2
```

z =

2	5	10
5	8	13
10	13	18

```
>> surf(x,y,z) % または、
```

```
>> surf(1:3,1:3,z)
```



これを応用し、`meshgrid` を全てのパターンの生成に利用する。

覆面算のルールより、**A,B** は、1 から 9 までのパターン、**C** は、0 から 9 までのパターンなので、まず、パターンを計算し、すべての解を計算し、`find` 関数により正解となるところを求める。この場合、`continue` 文がないが、計算速度は、十分速い。

ただし、この問題では重複がないため問題無いが、重複チェックする場合には、別のプログラムで行う必要がある。

```
tic
[A,B,C]=meshgrid(1:9,1:9,0:9);
kai = A*100+B*10+C-(B*10+A)-(B*10+B);
ind = find(kai==0);
[A(ind),B(ind),C(ind)]
toc
% 別解
tic
[A,B,C]=meshgrid(1:9,1:9,0:9);
A=A(:);B=B(:);C=C(:);
```

```
kai = A*100+B*10+C-(B*10+A)-(B*10+B);  
ind = find(kai==0);  
[A(ind),B(ind),C(ind)]  
toc
```

この方法は、スマートであるが、SEND+MORE=MONEY の場合には、大量のメモリを消費するため、使用には、注意すること。

Perms 関数を使う方法

Perms 関数を使うと、すべての組み合わせを出すことができる。

```
A=perms(0:9);  
% SEND+MORE=MONEY  
S=A(:,1);  
E=A(:,2);  
N=A(:,3);  
D=A(:,4);  
M=A(:,5);  
O=A(:,6);  
R=A(:,7);  
Y=A(:,8);  
% S and M is not zero  
ind=~((S==0)+(M==0));  
S=S(ind);  
E=E(ind);  
N=N(ind);  
D=D(ind);  
M=M(ind);  
O=O(ind);  
R=R(ind);  
Y=Y(ind);  
SEND=S*1000+E*100+N*10+D;  
MORE=M*1000+O*100+R*10+E;  
MONEY=M*10000+O*1000+N*100+E*10+Y;  
ind = find(SEND+MORE==MONEY)  
[ SEND(ind),MORE(ind),MONEY(ind)]  
toc
```

この場合、S と、M は、0 でないので、取り出した組み合わせから該当のインデックス番号以外のデータを取り出す。

あとは、計算をして find 関数により取り出すと出てくる。この場合には、2 つ同じ組み合わせのものが出てくる。

```
ind =      115661
          115662
ans =
          9567      1085      10652
          9567      1085      10652
経過時間は 0.731069 秒です。
```

SEND+10RE=1ONEY への適用

ここでは、計算量を減らすため、M=1 はわかっている場合の計算例を示す。meshgrid 関数は、3 次元までなので、ここでは、ndgrid 関数をつかう。

変数 S は、1 から 9 までであるが、1 はすでにわかっているため、2 からスタートしている。それ以外の変数 E,N,D,O,R,Y は、1 以外の数で、0 から 9 までの数であるため、ベクトルで、[0,2:9]という書き方をしている。

```
[S,E,N,D,O,R,Y]=ndgrid(2:9,[0,2:9],[0,2:9],[0,2:9],[0,2:9],[0,2:9],[0,2:9]);
S=S(:);E=E(:);N=N(:);D=D(:);
O=O(:);R=R(:);Y=Y(:);
SEND= S*1000+E*100+N*10+D;
MORE=1*1000+O*100+R*10+E;
MONEY=1*10000+O*1000+N*100+E*10+Y;
ind =(SEND+MORE==MONEY);
SENDMOREMONEY=[S(ind),E(ind),N(ind),D(ind),O(ind),R(ind),Y(ind)];
[row,col]=size(SENDMOREMONEY);
for i=1:row
    ok = 0;
    for j=[0,2:9]
        if(sum(SENDMOREMONEY(i,:)==j)> 1) ok = 1:end
        end
        if(ok == 0) SENDMOREMONEY(i,:),end
    end
end
```

変数 ind は、計算によりロジカル変数になるので、ロジカルインデックス機能を使って変数 S,E,N,D,O,R,Y から計算に合致する値のみを取り出している。

ロジカルインデックス機能を使わない場合には、

```
ind =find(SEND+MORE==MONEY);
```

としてもよい。(ただし、こちらの方が遅くなる。)

ここで出力された ind は、複数出てくる。理由は、全ての変数の重複チェックをしてないからである。そこで、重複チェックを行う。もともと量が少ないので、for 文によってやってみる。それぞれの ind のパターンで、0,2:9 までの数字を比較し、その数が 1 より大きい場合には、重複する数字があるとみなし、表示をしない。重複する数字がない場合、つまり、ok=0 のときの場合のみ表示する。ここでそれぞれ、S,E,N,D,O,R,Y の値が表示される。