Mathematicaプログラミング実習

古賀弘樹
筑波大学システム情報系

*Mathematica*プログラミングの基礎を学習し，リスト処理，グラフ理論，整数計画法等のプログラムを考察することで，必要に応じた簡単なプログラムを組めるようになることを目標とする．

1. Mathematicaの復習

■ リスト処理

【例題1.1】次の条件を満たすリストを作れ．

(1) 1から25までの整数

(2) 0から80までの3の倍数

(3) $\{n^2+1: n=1,2,..., 20\}$

(4) 20個の1から構成されるリスト

(5) ２重リスト{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}}

(6) 1以上100以下のランダムに発生させた20個の整数から構成されるリスト

【例題1.2】 リストa = {1, 2, 3, 4} , b = $\{x, x^2, x^3, x^4\}$ を用いて次のものを作れ．

(1) a の各要素に3を加えたリスト

(2) a の各要素を5 倍したリスト

(3) a の各要素を4乗したリスト

(4) $\{x, 2x^2, 3x^3, 4x^4\}$

(5) $\{1, 4x, 9x^2, 16x^3\}$

(6) $x + 2x + 3x^3 + 4x^4$

【例題1.3】リスト a = {1, {2, x}, 3, {4, {y, z}}, 5} に対して次の操作を行え．

(1) リストの第 1 要素を求めよ．

(2) リストの第 1 要素を除いたリストを求めよ．

(3) リストの最後の要素を除いたリストを求めよ．

(4) リストの第２要素の２番目の値を求めよ．

(5) リストの長さを求めよ．

(6) リストを平らにせよ．

【例題1.4】2重リストa = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}}に対して，以下の操作をせよ．

(1) 行列形式で表示せよ．

(2) 第2行目を求めよ．

(3) (3,2)成分を求めよ．

(4) 第2列を求めよ．

【例題1.5】a を-50以上50以下のランダムな20個の整数のリストとする．次の操作をせよ．

(1) 各要素を小さい順に並べよ．

(2) 各要素を大きい順に並べよ．

(3) 各要素を絶対値が小さい順に並べよ．

【演習１】a を-50以上50以下の一様ランダムな 10個の整数からなるリストとする． a の 3 個の要素の和の絶対値が最小になる組み合わせを探せ．

```
n = 10; k = 3;
a = RandomInteger[{-50,50},n];
b = Subsets[Range[1,n],{k}];
c = Table[{Abs[Total[a[[b[[i]]]]]],b[[i]]},{i,1,Length[b]}];
Sort[c,#1[[1]]<#2[[1]]&]
```

■ 関数の定義など

Map, 純関数等について復習する．

【例題1.6】リスト a = {1, 2, 3, 4, 5}に対して，各要素を 2乗して1 を加えたリストをいくつかの方法で作れ．

単純には

```
a^2 + 1
```

でよい．関数 f を定義し，Map を用いる方法もあり，こちらも応用が利く．関数の定義では:= が使われることに注意．

```
f[x_]:= x^2 + 1;  Map[f,a]
```

さらに，関数に名前をつけない純関数と呼ばれる方法を用いることもできる．

```
Map[#^2 + 1&,a]
```

b ={{1, 2}, {3, 4}, {5, 6}} に対して，b のレベル 1 の左側の要素に対して２乗して 1 を加えたリストが欲しければ，

```
Map[#[[1]]^2 + 1&,b]
```

とすればよい．b[[1]] = {1, 2}, b[[1, 2]] = 2 であることに注意する．

【演習２】1円硬貨をp, 5円硬貨をq, 10円硬貨をr, 50円硬貨をs, 100円硬貨をt, 500円硬貨をu と表す．リストcoin = {p, p, q, r, r, r, r, s, t, u, u} により財布の中に硬貨が何枚あるかを表しているとする．このとき，硬貨の合計がいくらであるかを求めるプログラムを作成せよ．Map, Count と内積を使え．

【例題1.7】Select 関数を用いて次の操作を行え．素数判定にはPrimeQ を使え．

(1) 1から100までのすべての素数からなるリストを作れ．

(2) 100000以下の完全数をすべて求めよ．なお整数 n が完全数(Perfect Number)であるとは，n のすべての約数の和が 2n に等しい（自分自身を除く約数の和が自分自身に等しい）と定義され，最も小さい完全数は 6 である．

2. グラフの描画とアルゴリズム

Mathematica はグラフの描画やアルゴリズムに関する様々なアルゴリズムを提供している．バージョン10より，Combinato-

rica パッケージに装備されている機能のほとんどがWolframシステムに実装されており，他の人が書いたプログラムを見るときは，バージョン10以降に対応したものか，古いバージョンに対応したものかを注意する必要がある．

【例題3.1】次のグラフを書け．また，頂点のラベル付け，辺のラベル付け，辺の重みの付け方等を調べよ．

(1) 完全グラフ K_5

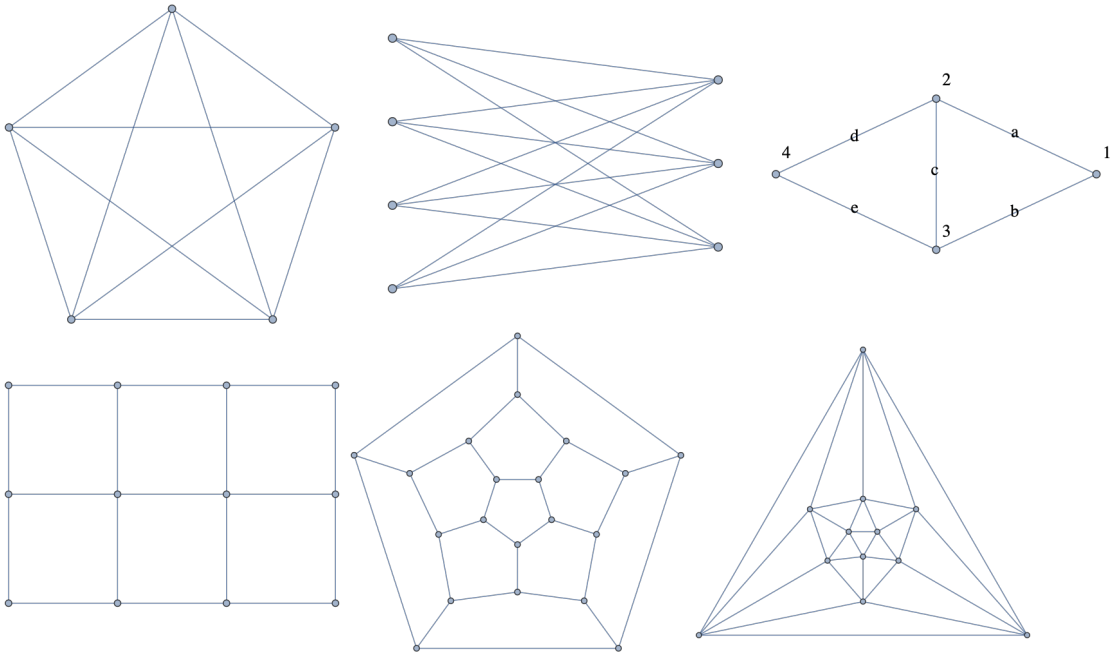
(2) 完全２部グラフ $K_{4,3}$

(3) 頂点集合 {1, 2, 3, 4}, 辺集合{{1, 2}, {1, 3}, {2,3}, {2, 4}, {3, 4}}のグラフ

(4) 3×4個の頂点をもつ格子グラフ

(5) 正12面体グラフ

(6) 正20面体グラフ

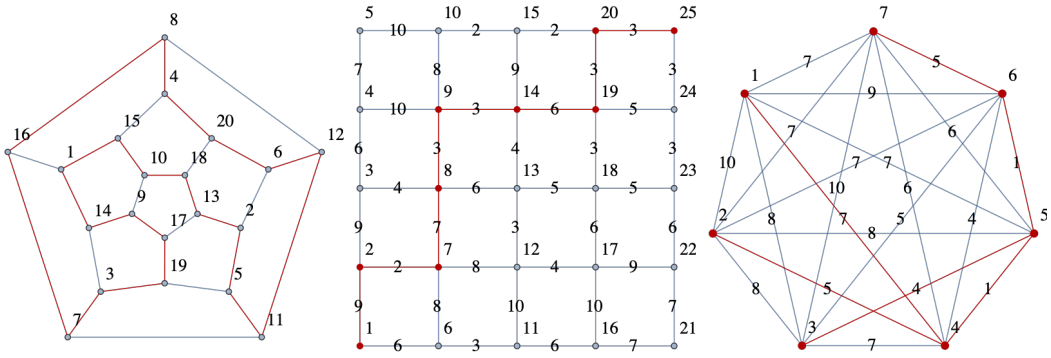


なお ⊷ は Esc + ue + Esc で表示できる．

【例題3.2】正12面体グラフにおいて，ハミルトン閉路を見い出して表示せよ．ハミルトン閉路とは，すべての頂点を１回ずつ通る閉路である．オイラー閉路は存在するか？

【例題3.3】5×5 の格子グラフにおいて，辺の重みを1以上10以下の整数の一様乱数で決定する．このとき，頂点１から頂点25に至る最小の重みの道を求めて表示せよ．

【例題3.4】完全グラフK_7 において，辺の重みを1以上10以下の整数の一様乱数で決定する．このグラフにおける最小全域木(minimum spanning tree)を求めて表示せよ．



3. 数学の問題に利用する

大学入試の問題を Mathematica を使って解いてみる．

【例題4.1】mを2015以下の正の整数とする．２項係数 $\binom{2015}{m}$ が偶数となる最小の m を求めよ．(2015年東大入試(改))

【例題4.2】3以上9999以下の奇数 a で $a^2 - a$ が10000で割り切れるものをすべて求めよ．　(2005年東大入試)

リストを扱う関数 SelectやPick を用いると簡単にプログラムできる．例題4.1 を「最小のm」でなくて「すべてのm」として考えると考えやすいかもしれない．例題4.2は「a が奇数」という制約を外した時の答も出そうと思えば出る．

フェルマーの小定理が成り立つことも簡単に確認できる．

【例題4.3】 p を任意に固定した素数であるとすると，任意の $a \in \{1, 2, ..., p\text{-}1\}$ に対して $a^{p-1} \equiv 1 \pmod p$ が成り立つ．ここに x ≡ y (mod p)は x - y を p で割った余りが 0 であることを意味する．これをいくつかの p に対して確認せよ．

数学の問題とはちょっと違うが，Mathematicaでは円周率πの近似値を何万桁も計算することができるので，指定する短い数列が円周率の何番目に現れるか調べることもできる．SequencePositionという関数の中に次のサンプルプログラムがあり，この中では314159 がπの何桁目に現れるかを調べてくれる．

```
digitsPi=First[RealDigits[N[Pi,10^7]]];
SequencePosition[digitsPi,{3,1,4,1,5,9}]
```

4. RSA暗号

RSA暗号として知られる公開鍵暗号では，大きな素数p, qを選んで $n = pq$ とし，$ed \equiv 1 \pmod{(p-1)(q-1)}$ を満たす2つの数 e,d を用いて，平文Mから暗号文Cを$C = M^e \bmod n$ と計算する．また，暗号文Cから平文Mを，$M = C^d \bmod n$ によって復号する．ここに，$ed \equiv 1 \pmod{(p-1)(q-1)}$ は，ed を $(p-1)(q-1)$で割ったあまりが 1 であるという意味であり，$M^e \bmod n$ はM^e を nで割った余りを表す．これを，Mathematica で実現してみよう．

まず，大きな素数p, q を2つ定める必要がある．実用的にはp, qは 2048ビット (680桁程度）と言われているが，時間がかかるので，5桁程度で考える．具体的には，適当に生成した５桁の数の素数判定を PrimeQ を用いて行えばよい．（当たり前ではあるが，偶数，3の倍数，5の倍数などは外しておく．実際，PrimeQ[22247] とPrimeQ[48539]はともにTrueとなるので，p=22247, q= 48539とする．このとき n = pq = 1079847133 になる．

次に，$ed \equiv 1 \pmod{(p-1)(q-1)}$ となる e, d を求める．eは素数を選ぶとよく，$e=65537$ としておく（$e=2^{2^4}+1$である）．d は *Mathematica* の*PowerMod*という関数を用いてd=*PowerMod[e,-1,(p-1)(q-1)] =196227113*と求まる．

ここまでくれば，Mを適当に選び，暗号化した後，復号すればもとに戻ることを確かめることは簡単である．Mを $n-1$ 以下の正整数からランダムに選び，cを計算して，もとに戻ることを確かめてみよう．

```
p = 22247; q = 48539; n = p q;
e = 65537;  d = PowerMod[e, -1, (p - 1) (q - 1)];
m = Table[RandomInteger[p q - 1], 10];    (* ランダムに平文を生成 *)
c = Map[PowerMod[#, e, n] &, m];          (* 平文を暗号化 *)
Map[PowerMod[#, d, n] &, c] - m      (* 暗号文を復号，mと一致すれば 0 *)
```

5. 数独(ナンプレ)を解く

数独（ナンプレ）は整数計画法を用いて解けることが知られている．整数計画法は，変数に整数条件を課した線形計画問題と思ってよく，一般には解くのが大変であるが，9×9 のナンプレであれば答がすぐに見つかる．

整数計画法では（線形計画法でも）LinearOptimization という関数を用いる．たとえば

```
minimize       x1 + x2 + x3 + x4
subject to     x1 + x2 == 1
               x2 + x3 == 1
               x3 + x4 == 1
               x1, x2, x3, x4 ≥ 0, Integers
```

という問題を考えてみる．x_1, x_2, x_3, x_4が非負整数であるという制約があるので，$x_1 + x_2 = 1$は，x_1, x_2 のどちらか一方が　1,　他方が　0　と考えることができる．少し考えればわかるように，$(x_1, x_2, x_3, x_4) = (1, 0, 1, 0)$ or $(0, 1, 0, 1)$が最適解になる．Mathematica では，次のように書けば良い．

```
m = {{1,1,0,0},{0,1,1,0},{0,0,1,1}};
LinearOptimization[Total[v], m. v == {1,1,1}, v ∈ Vectors[4, NonNegativeIntegers]]
```

このプログラムにより，最大値を与える{v→{1,0,1,0}}が求まる．

数独も，整数計画法の問題に定式化をすることができる．いまi, jをマスの位置を示す番号(1≤i, j≤9) とし，k ∈{1, 2, ..., 9} をマスの中に入れる値とする．変数x_{ijk} を次で定義する．

$$x_{ijk} = \begin{cases} 1, & (i, j) \text{ のマスの値がk のとき} \\ 0, & \text{それ以外のとき} \end{cases}$$

各(i, j)のマスに対して，1から9までの数字が必ず1つ入るので,

$$\sum_{k=1}^{9} x_{ijk} = 1 \quad \text{for all } i, j \in \{1, 2, \ldots, 9\}$$

が成り立つ．各行には必ず1から9の値が 1 つずつ入るので,

$$\sum_{j=1}^{9} x_{ijk} = 1 \quad \text{for all } i, k \in \{1, 2, \ldots, 9\}$$

が成り立つ．同様に，各列にも必ず1から9の値が1つずつ入るので

$$\sum_{i=1}^{9} x_{ijk} = 1 \quad \text{for all } j, k \in \{1, 2, \ldots, 9\}$$

がいえる．さらに，3×3の 9 個の位置にも必ず1から9の値が1つずつ入るので，各(s, t)=(1, 1), (1, 4), (1, 7), (4, 1), (4, 4), (4, 7), (7, 1), (7, 4), (7, 7) に対して

$$\sum_{i=0}^{2} \sum_{j=0}^{2} x_{s+i,t+j,k} = 1 \quad \text{for all } k \in \{1, 2, \ldots, 9\}$$

が成り立つ．この他，数独の場合は，さらに最初に与えられる数字（初期値）があるので，例えば(1,3)のマスが　5なら，$x_{135} = 1$ という条件を与える．

数独の場合，変数の数が9×9×9=729個，制約の数が，数独のルールに由来するものが81×4=324個で，これに初期値に由来する制約 (多くの場合20〜30個程度) が加わる．サンプルコードをつけておく．

```
n = 9;    hn = 3; (* hn = SQRT(n) *)
sqn = n n;  cbn = n n n;
(*                    初期配置                          *)
init={{1,1,4},{1,5,1},{2,3,8},{2,7,2},{3,1,7},{3,3,9},{3,4,4},{3,5,8},{3,8,3},{4,2,4},{4,8,
7},{4,9,5},{5,6,2},{5,9,8},{6,2,3},{6,4,7},{7,4,8},{7,7,1},{7,9,2},{8,5,4},{8,8,9},{9,1,6},
{    9    ,    8    ,    8    }    }    ;
zero=Table[0,{i,1,cbn}];
f[{i_,j_,k_}]:= sqn (i-1) + n (j-1) +k;
one[x_]:=             1;

(* 制約条件１:各マスに数字は１つ *)
tmp1 = Flatten[Table[{f[{i,j,k}]},{i,1,n},{j,1,n},{k,1,n}],1];
constraint1=Table[MapAt[one,zero,tmp1[[i]]],{i,1,sqn}];

(* 制約条件２:各行に数字は１つ *)
tmp2= Flatten[Table[{f[{i,j,k}]},{k,1,n},{i,1,n},{j,1,n}],1];
constraint2=Table[MapAt[one,zero,tmp2[[i]]],{i,1,sqn}];
```

```
(* 制約条件3:各列に数字は1つ *)
tmp3= Flatten[Table[{f[{i,j,k}]},{k,1,n},{j,1,n},{i,1,n}],1];
constraint3=Table[MapAt[one,zero,tmp3[[i]]],{i,1,sqn}];

(* 制約条件4: 3*3 ボックス9つに関する制約 *)
cp={{1,1},{1,4},{1,7},{4,1},{4,4},{4,7},{7,1},{7,4},{7,7}};
box[{x_,y_}]:= Table[Flatten[Table[{f[{x+i,y+j,k}]},{i,0,hn-1},{j,0,hn-1}],1],{k,1,n}];
tmp4=Flatten[Map[box,cp],1];
constraint4=Table[MapAt[one,zero,tmp4[[i]]],{i,1,Length[tmp4]}];

(* 初期配置を制約条件にする *)
tmp5=Partition[Partition[Map[f,init],1],1];
constraint5       =       Table[MapAt[one,zero,tmp5[[i]]],{i,1,Length[tmp5]}];

(* 制約条件を表す行列とその右辺を作る*)
m = Join[constraint1,constraint2,constraint3,constraint4,constraint5];
allone              =             Table[1,Length[m]];

(*整数計画法を解く *)
sol = Flatten[LinearOptimization[Total[v],m.v==allone,v ∈ Vectors[cbn, NonNegativeInte-
gers], "PrimalMinimizer"]];

(* 解の確認 *)
psol=Partition[Flatten[sol],n];
Partition[Flatten[Table[Position[psol[[i]],1],{i,1,Length[psol]}]],n]
```

6. 課題

これまでのことから，様々な問題で Mathematica を使うことができることがわかり，プログラミングのスキルも身についていると思う．残った時間で，自分で簡単なMathematicaプログラミングで解ける問題を見つけてレポートにして提出すること．

以下の注意をよく読み，

- 問題，Mathematica プログラム，解答を明記する．
- 問題は，極力Mathematica 内の文書内にないものにする．Mathematica 内のサンプルプログラムを使うときは自分なりに十分検討すること．
- プログラムで何を計算しているか，適宜コメントを入れるなど，日本語の説明をつけること．
- 問題の面白さ・斬新さ，Mathematica を用いることで解が導ける鮮やかさ，解の意外さがわかるもの，または関数のオプションを使ってうまく表示されているものは高く評価します．
- 7月21日(水) 23:55 までに pdf ファイルをmanaba に提出する．

参考文献

A. 白石修二，「例題で学ぶMathematica [基礎プログラム編]」，森北出版，2000.
B. 榊原進，「はやわかりMathematica（第３版）」，共立出版，2010.
C. S. Mangano (松田裕幸訳），「Mathematicaクックブック」，オライリー・ジャパン，2011.
D. S. Skiena （植野義明訳），「Mathematica組み合わせ論とグラフ理論」，Addison-Wesley Toppan, 1992.