

自主課題研究

数理処理ソフトを用いた数学探求

担当： 佐藤 卓治

テーマ『再起的関数』

工学部 情報システム工学科 3年

名列番号： 043

氏名： 津田 志朗

学籍番号： 0308060442

0. 自主課題研究における背景

0-1 研究目的

数値処理ソフト(Mathematica)を用いて再帰的関数を研究し、興味深い関係を模索・発見する。Mathematica の使い方を復習しているうちに、自らの値を持って次の値を求める再帰的関数に興味を持ち、さらには調べていくうちに様々な面白い関係が見いだせそうであったため、再帰的関数をテーマに調査をはじめた。

0-2 再帰的関数とは

再帰的関数というのは関数の定義の中にその関数自身が参照されていることをいう。数学では再帰的関数とはいわずに一般的に漸化式という。

よって、主に数列の漸化式というものを取り入れて調査を行いたいと思う。

1. hanoi の塔

1-1 hanoi の塔とは

3本の棒 a,b,cがあり,そのうちの一つの棒に大きさがそれぞれ異なる輪が大きさの順に並べられている.これを塔と呼ぶことにする.

今、この塔を別の棒に移したい。ただし、この時のルールが3つあり、

1. 輪は一つずつしか動かせない。
2. 大きな輪は小さな輪の上に乗せることはできない。
3. 輪を棒 A,B,C 以外のところに置くことはできない。

このルールに乗っ取り、塔を移し変える。これが hanoi の塔の問題である。

1-2 研究

輪を二つで考えると、

- (1)上の輪が入っていた棒から余っている棒へ移動させる
- (2)下の輪を目的の棒へ移動させる
- (3)上の輪を目的の棒へ移動させる

この三つの単純な作業でできる。

これを高さ n の塔で考える場合も、1 から n-1 の輪を1まとまりの輪と考えると、二つの輪で考える場合と同じことになる。

よって、このことから関数 hanoi を用いて再帰的なプログラムを作成すると、

```

hanoi[1,from_,to_,spare_] := StringJoin[from, " ", "→", " ", to]
hanoi[n_,from_,to_,spare_] :=
  Flatten[{hanoi[n-1,from,spare,to],hanoi[1,from,to,spare],
    hanoi[n-1,spare,to,from]}]

```

これは、上記の(1)から(3)の動作をさせるプログラムである。ただし、棒 a から棒 b へ移動させることにする。

1-2-1 n 段の塔に対する輪の移動回数

n 段の塔があった場合、どれだけ輪を移動させなければいけないのか考える。

1 段 → 1 回、2 段 → 3 回、3 段 → 7 回、4 段 → 15 回、5 段 → 31 回移動させていることが分かる。これらの数字をよく見ると、+1 すると 2^n の形になっているので、移動させる回数を $H(n)$ とおくと、

$$H(n) = 2^n + 1$$

が成り立つと予想できる。

これを式で証明する。

輪を移動させる際に、 $H(n-1)$ 回かけて $n-1$ 段の塔を A から C に移し、n 段目を A から B へ移し、そのあとに $H(n-1)$ 回かけて $n-1$ 段を C から B に移す作業をするので、 $H(n)$ を式で表すと、

$$\begin{aligned}
 H(n) &= H(n-1) + 1 + H(n-1) \\
 &= 2H(n-1) + 1 \\
 H(1) &= 1
 \end{aligned}$$

この漸化式を計算すると、

$$\begin{aligned}
 H(n) + 1 &= 2^n \\
 H(n) &= 2^n - 1
 \end{aligned}$$

よって n 段の塔の移動回数は $2^n - 1$ 回であることが分かる。

これを関数 H で次のように定義すると、簡単に移動回数を求めることができる。

$$H[n_] := 2^n - 1$$

1-2-2 塔を移動させるのにかかる時間

リュカ(Edouard Lucas)の作り話に次のような話がある。

インドのベレナスにある大寺院にダイヤモンドの針が 3 本立った板があり、神はその 1 本に 64 枚の純金の円盤をはめた。昼夜の別なく、バラモン僧たちはそこにやってきて、それを別の針に移し替える作業に専念している。そして移し替えが完了したとき、寺院もバラモン僧たちも崩壊し、この世が終るのである。

ここで、1 枚の輪を移動させるのに 1 秒かかるとして、この世の終りまでに何年かかるのか計算する。

先ほど求めた関数 H を用いて 64 段の塔を移動させるのに何回輪を移動させるのかを求めると、18446744073709551615 回移動させることが分かった。これを 1 年が 365 日として計算すると約 5850 億年かかると計算できる。まだまだこの世の終わりを心配する必要はないようである。

では次に実習室のコンピュータで 64 段積むのにどのくらいかかるのか計算してみる。

まず、1 回輪を移動させるのにかかる時間を考える。実際に何回も計算させてその平均をとると、大体 1 つ輪を移動させるのに 0.000027 秒かかることが分かった。

同様に同じ回数だけ輪を移動させると 64 段の塔を移動させるのに約 1579 万年かかることが分かる。世界が滅びるのは本当にまだまだ先のことである。

2. ユークリッドの互除法

2-1 ユークリッドの互除法とは

ユークリッドの互除法というのは、2 つの数の最大公約数を効率よく求める方法のことである。今、2 つの数 m, n の最大公約数を求めるとし、その値を $\text{GCD}(m, n)$ と表す。また、 m, n の剰余を l とするとき、次の式が成り立つ。(証明は省略)

$$\text{GCD}(m, n) = \text{GCD}(n, l)$$

$m > n > l$ ということに気をつけると、 m, n に比べて n, l の方が値が小さいので計算が簡単である。それでも値が大きいときは n を l で割って同様の操作を行えばよい。そして

$$\text{GCD}(x, 0), x > 0$$

の形になるまで続けると、 $\text{GCD}(x, 0) = x$ となり、これで $\text{GCD}(m, n)$ が求められる。

2-2 研究

Mathematica により、ユークリッドの互除法のプログラムを作ると以下のようになる。

```
myGCD[m_Integer, 0] := m
```

```
myGCD[m_Integer, n_Integer] := myGCD[n, Mod[m, n]]
```

*_Integer は整数を表す。Mod[x, y] は変数 x, y の剰余を表す。

また、Mathematica に搭載されている関数を用いて求めることもできる。

ExtendedGCD[a, b] を入力すると {d, {s, t}} が出力される。(d: 最大公約数)

3. フィボナッチ数列

3-1 フィボナッチ数列とは

初項と第2項が1, それ以降の項は前の2項の加算によって求められる数列を**フィボナッチ数列**という。

漸化式で表すと、

$$\begin{aligned}a_1 &= a_2 = 1 \\ a_n &= a_{n-1} + a_{n-2} \quad (n \geq 2)\end{aligned}$$

3-2 研究

関数 **fibonacci** においてフィボナッチ数列を Mathematica にて以下のように定義する。

`fibonacci[1] = fibonacci[2] = 1;`

`fibonacci[n_]:= fibonacci[n-2] + fibonacci[n-1]`

3-2-1 ビネの公式 (フィボナッチ数列の一般項)

フィボナッチ数列の一般項は**ビネの公式**として知られている。以下の通りとなる。

$$a_n = \frac{1}{\sqrt{5}}(\alpha^n - \beta^n) \quad \text{ただし } \alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$$

このビネの公式を関数 `a` を用いて Mathematica にて定義をする。

$$\alpha = \frac{1+\sqrt{5}}{2};$$

$$\beta = \frac{1-\sqrt{5}}{2};$$

$$a[n_]:= N[\frac{1}{\sqrt{5}}(\alpha^n - \beta^n),1000]// Round$$

*`N` は近似値を求める関数で,今は 1000 桁まで計算するよう設定してある

*`Round` は小数点以下を切り捨ててくれる関数

3-3 フィボナッチ数の性質

3-3-1 性質(1)フィボナッチ数の隣り合った数は互いに素である。(以下、証明は省略)

この性質から、フィボナッチ数列を求める動作というのは、ユークリッドの互除法を行っているのと同じ動作であるということが分かる。隣り合った2つのフィボナッチ数8と13をユークリッドの互除法にかけて最大公約数を求めてみる。

$$13 \div 8 = 1 \text{ あまり } 5 \rightarrow 13 - 8 = 5$$

$$8 \div 5 = 1 \text{ あまり } 3 \rightarrow 8 - 5 = 3$$

$$5 \div 3 = 1 \text{ あまり } 2 \rightarrow 5 - 3 = 2$$

隣り合うフィボナッチ数が互いに素であるので商は常に1となる。よって右の式に変形することができ、これは $a_n = a_{n-1} + a_{n-2}$ の式になっている。

3-3-2 性質(2)

フィボナッチ数を順に2乗してn項まで足すと、n項とn+1項の積と等しくなる。

$$a_1^2 + a_2^2 + a_3^2 + \dots + a_n^2 = a_n \times a_{n+1}$$

$$\sum_{k=1}^n a_k^2 = a_n \times a_{n+1}$$

3-3-3 性質(3)

フィボナッチ数列を順にn項まで足すと、n+2項から1引いたものと値が等しくなる。

$$a_1 + a_2 + a_3 + \dots + a_n = a_{n+2} - 1$$

$$\sum_{k=1}^n a_k = a_{n+2} - 1$$

3-3-4 性質(4)

性質(4-1)

フィボナッチ数列の奇数項だけn項まで足した時、n+1項と同じ値になる。

$$a_1 + a_3 + a_5 + \dots + a_n = a_{n+1} \quad \text{ただし } n = 2m - 1 \text{ とする。}$$

$$\sum_{m=1}^n a_{2m-1} = a_{n+1}$$

性質(4-2)

偶数項をn項まで足した時、n+1項から1引いた値と等しくなる。

$$a_2 + a_4 + a_6 + \dots + a_n = a_{n+1} - 1 \quad \text{ただし } n = 2m \text{ とする。}$$

$$\sum_{m=1}^n a_{2m} = a_{n+1} - 1$$

3-3-5 性質(5)

3つ続いたフィボナッチ数の外2つをかけたものから、真ん中の値を2乗したものを引くと、1と-1が交互に現れる。

$$a_n \times a_{n+2} - a_{n+1}^2 = (-1)^{n-1}$$

4. リュカ数

4-1 リュカ数とは

初項が1、第2項が3、それ以降の項は前の2項の加算によって求まる数列をリュカ数という。フィボナッチ数の第2項が3に置き換わっただけの数列である。

漸化式で表すと

$$b_1 = 1, b_2 = 3$$

$$b_n = b_{n-1} + b_{n-2} \quad (n \geq 2)$$

となる。

4-2 研究

Lucas[]という関数で定義する。

Lucas[1]=1;

Lucas[2]=3;

Lucas[n_]:=Lucas[n]=Lucas[n-1]+Lucas[n-2]

リュカ数はフィボナッチ数と作りが同じなので、隣り合う2つの数はまた互いに素である。

4-2-1 リュカ数の一般項

フィボナッチ数の時はビネの公式という一般項を求める公式があった。リュカ数において、特性方程式を用いて求めると以下の式が出てくる。(3項間の漸化式の一般項)

$$b_n = \alpha^n + \beta^n \quad \text{ただし、} \quad \alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$$

この一般項をbという関数を用いて定義する。

b[1]=1;

b[2]=3;

b[n_]:=b[n-1]+b[n-2]

4-2-2 特殊な数

フィボナッチ数とリュカ数、2つの数の一般項に着目してみると非常によく似た形をしているのが分かる。

$$a_n = \frac{1}{\sqrt{5}}(\alpha^n - \beta^n) \quad \text{ただし} \quad \alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$$

$$b_n = \alpha^n + \beta^n \quad \text{ただし} \quad \alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$$

この2つの一般項より β^n を削除した $c_n = \sqrt{5}a_n + b_n$ という数について考えてみる。

まず、一般項について考える。

$$c_n = \sqrt{5}a_n + b_n$$

$$= \sqrt{5} \cdot \frac{1}{\sqrt{5}}(\alpha^n - \beta^n) + \alpha^n + \beta^n = 2\alpha^n = 2\alpha \cdot \alpha^n$$

この数列 c_n は、初項が 2α で公比が α の等比数列であることが分かる。

では、この数列にはフィボナッチ数列やリュカ数のような再帰性があるのだろうか。以下の方法で確かめる。

$c[n_] := \sqrt{5} \text{ fibo}[n] + \text{Lucas}[n]$ (関数 c で数列 c_n を定義する。)

$\text{Table}[c[n]/\text{Expand}, \{n, 1, 10\}]$ (関数 c を第 10 項まで表示させる。)

$\{1 + \sqrt{5}, 3 + \sqrt{5}, 4 + 2\sqrt{5}, 7 + 3\sqrt{5}, 11 + 5\sqrt{5}, 18 + 8\sqrt{5}, 29 + 13\sqrt{5}, 47 + 21\sqrt{5}, 76 + 34\sqrt{5}, 123 + 55\sqrt{5}\}$

もし数列 c_n に再帰的な性質があるのならば $c_n = c_{n+2} - c_{n+1}$ という形をとるので、

$\text{Table}[\text{orig}[n + 1] - \text{orig}[n], \{n, 2, 11\}]$

$\{1 + \sqrt{5}, 3 + \sqrt{5}, 4 + 2\sqrt{5}, 7 + 3\sqrt{5}, 11 + 5\sqrt{5}, 18 + 8\sqrt{5}, 29 + 13\sqrt{5}, 47 + 21\sqrt{5}, 76 + 34\sqrt{5}, 123 + 55\sqrt{5}\}$

よって数列 c_n 再帰的な性質を持ち、かつ等比数列でもある特殊な数であることが分かった。またこのことより、再帰的な数同士の一次結合もまた再帰的な数であるということも証明された。

5. ソモスの数列

初項から第 4 項までの値が 1 で、それ以降が次の漸化式を持つ数列をソモスの数列という。

$$d_4 = d_3 = d_2 = d_1 = 1$$

$$d_n \times d_{n+4} = d_{n+3} \times d_{n+1} + d_{n+2}^2$$

実際に Mathematica で定義をして、どのような値をとるのか見てみると

$\text{sms}[1] = \text{sms}[2] = \text{sms}[3] = \text{sms}[4] = 1$

$\text{sms}[n_] := \text{sms}[n] = (\text{sms}[n - 3] \text{sms}[n - 1] + \text{sms}[n - 2]^2) / \text{sms}[n - 4]$

$\text{Table}[\text{sms}[n], \{n, 1, 20\}]$ (第 20 項まで表示させる。)

$\{1, 1, 1, 1, 2, 3, 7, 23, 59, 314, 1529, 8209, 83313, 620297, 7869898, 126742987, 1687054711, 47301104551, 1123424582771, 32606721084786\}$

この数列は漸化式の形とは裏腹に、任意の n に対して全ての d_n は整数になるという特徴がある。

6. コンウェイの数列

まず,任意の2つの整数を置き,次に後の数に1加えて前の数で割り,その解を後の数の次に置くという作業をして生まれる数列をコンウェイの数列という.この数列を定義してどのような値をとるのか見てみる。関数 **con** で定義する。

$con[1] = con[2] = 1;$

$$con[n_]:=con[n] = \frac{con[n-1]+1}{con[n-2]}$$

Table[con[n],{n,1,20}]

{1,1,2,3,2,1,1,2,3,2,1,1,2,3,2,1,1,2,3,2}

このコンウェイの数列は,5 回ごとに同じ値の列が繰り返されるという非常に興味深い性質を持っている。

7. ホフスタッター数列

ホフスタッター数列とは、次の4つの性質を持つ数列のことをいう。

- (1)n を引数とする再帰式を持つ。
- (2)初期値を有限個与えられる。
- (3)数列の値はすべて自然数の値をとる。
- (4)グラフにすると一次直線など、単純な多項式で表される曲線によっておおむね近似されるもの。

(注) ただし、再帰式や初期値の設定によっては「無限数列」にならなかったり、「すべての自然数の引数で自然数の値」をとらなかったりする場合もある。

7-1 ホフスタッター-G 数列

次の漸化式で表される数列を**ホフスタッター-G 数列**という。

$$g_1 = 1$$

$$n - g_{(g_{n-1})} \qquad n \geq 2$$

実際に定義をして表とグラフにしてみると,(関数 **G** で定義する。)

G[1]=1;

G[n_]:=G[n]=n-G[G[n-1]]

Table[G[n],{n,1,20}] (第 20 項まで表示させる。)

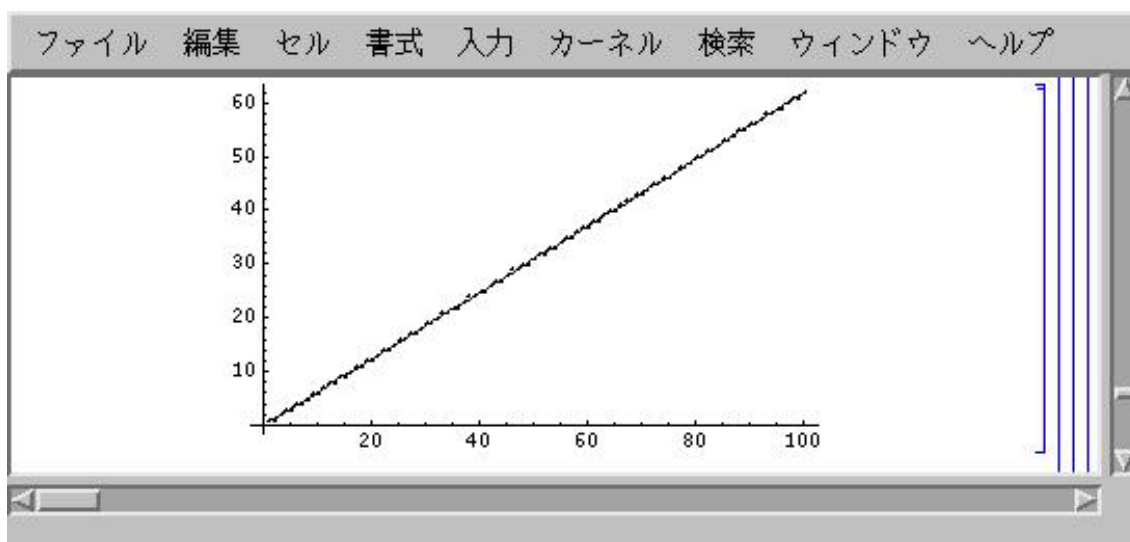
{1,1,2,3,3,4,4,5,6,6,7,8,8,9,9,10,11,11,12,12}

またこの数列は $f(x) = \frac{x \cdot (-1 + \sqrt{5})}{2}$ で近似されるので一緒にプロットを行う。

```
Table[G[n],{n,1,100}]/ListPlot; ( 第 100 項までをグラフにする。)
```

```
Plot[(-1+  $\sqrt{5}$ ) n/2, {n, 1, 100});
```

```
Show[%%,%]
```



近似されている様子がよく分かる。

7-2 ホフスタッタ-Q 数列

次の漸化式で表される数列を **ホフスタッタ-Q 数列** という。

$$q_1 = q_2 = 1$$

$$q_n = q_{(n-q_{n-2})} + q_{(n-q_{n-1})}$$

この数列においても Mathematica で定義を行い、表とグラフを作成する。

```
Q[1]=Q[2]=1;
```

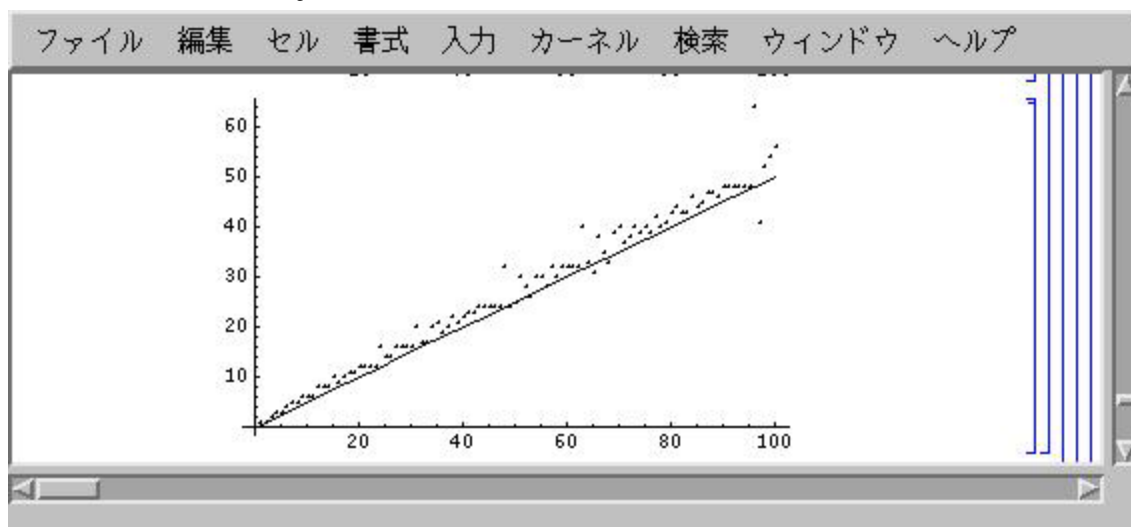
```
Q[n_]:=Q[n]=Q[n-Q[n-1]]+Q[n-Q[n-2]]
```

```
Table[Q[n],{n,1,20}]
```

```
{1,1,2,3,3,4,5,5,6,6,6,8,8,8,10,9,10,11,11,12}
```

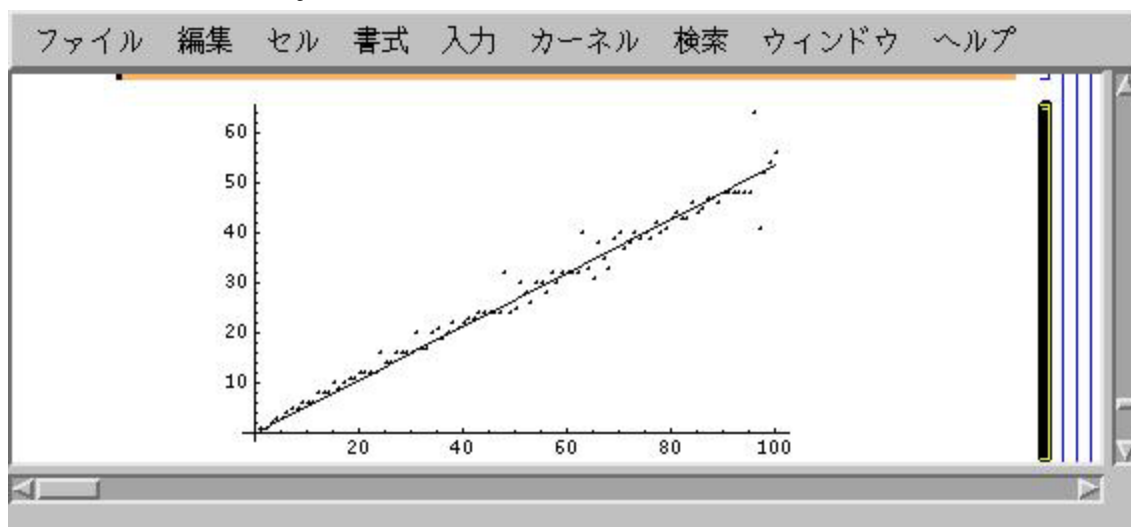
また、この数列は関数 $f(x) = \frac{x}{2}$ と近似されるので、両方表示させる。

```
Show[ListPlot[Table[Q[n], {n, 1, 100}]], Plot[n/2, {n, 1, 100}]]
```



ややズレているので、 $f(x) = \frac{8n}{15}$ と比べてみる。

```
Show[ListPlot[Table[Q[n], {n, 1, 100}]], Plot[8n/15, {n, 1, 100}]]
```



プロット点を増やして比較してみても、より近い近似式ではないかと予想できる。

7-3 ホフスタッターコンウェイの 10,000 ドル数列

次のような漸化式で表される数列をホフスタッター-コンウェイの 10,000 ドル数列という。

$$\$_1 = \$_2 = 1$$

$$\$_n = \$_{(\$_{n-1})} + \$_{(\$_{n-2})} \quad n \geq 2$$

関数 \$ として定義を行う。

```
$[1]=$[2]=1;
```

```
$[n_]:= $[n]=$[$[n-1]]+$[n-$[n-1]]
```

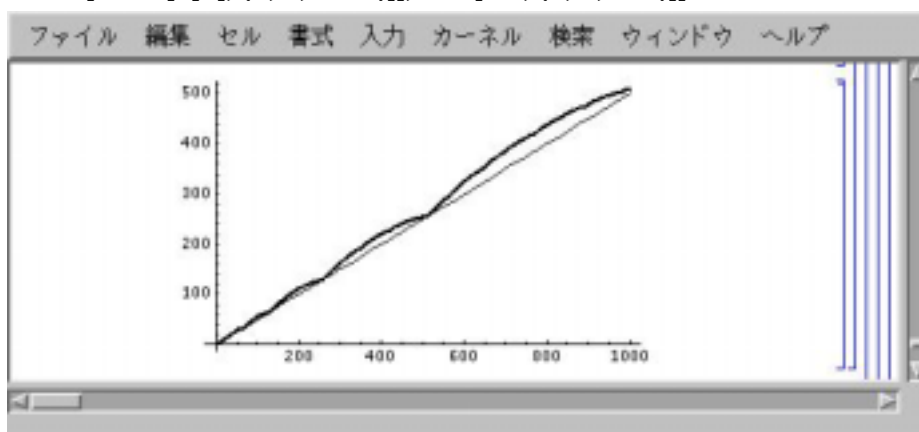
表にしてみると,

```
Table[$[n],{n,1,20}]
```

```
{1,1,2,2,3,4,4,5,6,7,7,8,8,8,9,10,11,12}
```

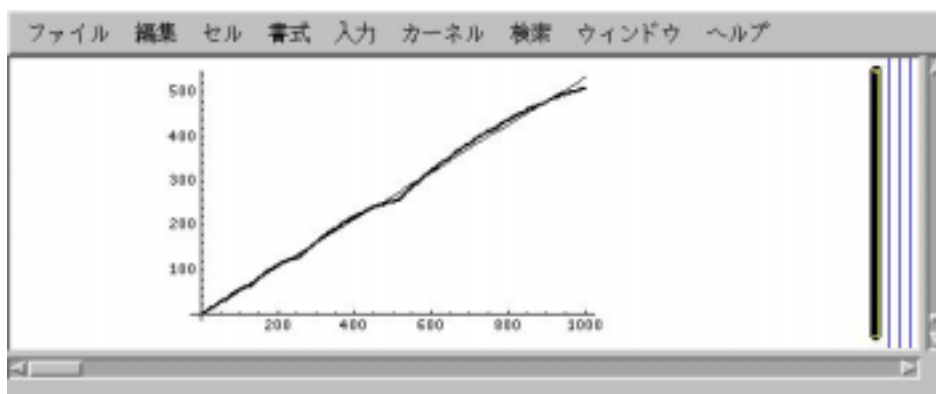
この数列も $f(x) = \frac{x}{2}$ で近似されるので両方プロットしてみる。

```
Show[ListPlot[Table[$[n], {n, 1, 1000}]], Plot[n/2, {n, 1, 100}]]
```



先ほどと同様に、少しズレているので $f(x) = \frac{8x}{15}$ と比べてみる。

```
Show[ListPlot[Table[$[n], {n, 1, 1000}]], Plot[(8n)/15, {n, 1, 1000}]]
```



8. まとめ

今回『再帰的関数』をテーマに掲げ、調査・模索を行ってきた。再帰的関数とはいっても、そのほとんどの対象を数列の漸化式を中心にしてきた訳であるが、新たな発見が多かった。その要点をまとめてみる。

1. hanoi の塔の問題は再帰的に定義することで一般化することができる。(1-2)
2. フィボナッチ数を求める動作はユークリッドの互除法と同じ動作である。(3-3-1)
3. フィボナッチ数のような再帰的性質を持ち、かつ等比数列の関係にある数が存在する。(4-4-2)
4. フィボナッチ数列のような再帰的な数同士の一時結合もまた再帰的な数である。(4-4-2)
5. 初期値を加えれば、あとは5つずつ同じ数が繰り返されるコンウェイ数列が存在する。(6)
6. 調べるだけで聞いたことのない数列が多数存在する。

この研究をするに当たって、以上6点が特に興味を持った部分である。その中でも4については自分でそのような数をたまたまではあるが発見できたので、また1つ数学という世界に興味を覚えた。

今後、また再帰的関数だけにとどまらず、数学の面白さ奥深さについて調査を行っていきたいと思う次第である。

参考文献

「実験数学入門」 山本 芳彦著

「Mathematica ハンドブック」 木村 広著