フィボナッチ数列は2進数でも美しいのか

京都教育大学附属京都小中学校　9年　吉田桃子

1.動機

フィボナッチ数列は自然界に多く存在する数である。例えば、花びらの数やひまわりの種の数などに現れている。その一方で、2進数は主にコンピュータに使われるものである。一見、対象的な2つに見えるかもしれないが、私達が普段使っている10進数と2進数は使う数字の幅が違うだけであって、根本的には変わらないはずである。ならば10進数で表されたフィボナッチ数列を2進数にしたときに、何か法則が現れたり、規則性があったりするのではないか、と考えた。
そこで、今回フィボナッチ数列を2進数に変換して調べてみることにした。

2.研究の方法

①2進数に変換したフィボナッチ数列を用意する
・インターネットやエクセルなどの変換器では数が大きすぎて十分な量を変換できなかったので、自分で計算用のプログラムを作って計算した。*1
・下記記載の方法を使って求めた。*2
・フィボナッチ数列は無限に続くので、今回は100番目までを使用した。*3

ⅰ もとの10進数を2で割り、余りを求める。
ⅱ 商が0になるまで割り続ける。
ⅲ 余りを出た順番の逆で並べる。

（例）14の場合
14÷2＝7　・・・0
7÷2＝3　・・・1
3÷2＝1　・・・1
1÷2＝0　・・・1
⇒14を2進数で表すと「1110」となる。

②0を「○」、1を「●」にとして表示する。
・0と1のままだと分かりにくいので、一覧性があるようにした。

③変換した○と●の表から読み取れることを考える。

3.予想

現れる規則性や法則は、

①桁の数
②○と●の数の比
③各位の数の合計
④縦の規則性

のようなものだと予想した。
特に数学的な根拠があるというわけではないが、フィボナッチ数の規則性や自然界での存在の大きさから①～④のようなものが現れるだろうと推測した。

4.結果

番目	フィボナッチ数列	2進数
1	1	●
2	1	●
3	2	●○
4	3	●●
5	5	●○●
6	8	●○○○
7	13	●●○●
8	21	●○●○●
9	34	●○○○●○
10	55	●●○●●●
11	89	●○●●○○●
12	144	●○○●○○○○
13	233	●●●○●○○●
14	377	●○●●●●○○●
15	610	●○○●●○○○●○
16	987	●●●●○●●○●●
17	1597	●●○○○●●●●○●
18	2584	●○●○○○○●●○○○
19	4181	●○○○○○●○●○●○●
20	6765	●●○●○○●●○●●○●
21	10946	●○●○●○●●○○○○●○
22	17711	●○○○●○●○○●○●●●●
23	28657	●●○●●●●●●●●○○○●
24	46368	●○●●○●○●○○●○○○○○
25	75025	●○○●○○●○●○○○●○○○●
26	121393	●●●○●●○●○○○●●○○○●
27	196418	●○●●●●●●●●○●○○○○●○
28	317811	●○○●●○●●○○●○●●●○○●●
29	514229	●●●●●○●●○○○●○●●○●○●
30	832040	●●○○●○●●○○●○○○●○●○○○
31	1346269	[illegible]
32	2178309	[illegible]
33	3524578	[illegible]
34	5702887	[illegible]
35	9227465	[illegible]
36	14930352	[illegible]
37	24157817	[illegible]
38	39088169	[illegible]
39	63245986	[illegible]
40	102334155	[illegible]
41	165580141	[illegible]
42	267914296	[illegible]
43	433494437	[illegible]
44	701408733	[illegible]
45	1134903170	[illegible]
46	1836311903	[illegible]
47	2971215073	[illegible]
48	4807526976	[illegible]
49	7778742049	[illegible]
50	12586269025	[illegible]
51	20365011074	[illegible]
52	32951280099	[illegible]
53	53316291173	[illegible]
54	86267571272	[illegible]
55	139583862445	[illegible]
56	225851433717	[illegible]
57	365435296162	[illegible]
58	591286729879	[illegible]
59	956722026041	[illegible]
60	1548008755920	[illegible]
61	2504730781961	[illegible]
62	4052739537881	[illegible]
63	6557470319842	[illegible]
64	10610209857723	[illegible]
65	17167680177565	[illegible]
66	27777890035288	[illegible]
67	44945570212853	[illegible]
68	72723460248141	[illegible]
69	117669030460994	[illegible]
70	190392490709135	[illegible]
71	308061521170129	[illegible]
72	498454011879264	[illegible]
73	806515533049393	[illegible]
74	1304969544928657	[illegible]
75	2111485077978050	[illegible]
76	3416454622906707	[illegible]
77	5527939700884757	[illegible]
78	8944394323791464	[illegible]
79	1.44723340246762E+016	[illegible]
80	2.34167283484677E+016	[illegible]
81	3.78890623731439E+016	[illegible]
82	6.13057907216116E+016	[illegible]
83	9.91948530947555E+016	[illegible]
84	1.60500643816367E+017	[illegible]
85	2.59695496911123E+017	[illegible]
86	4.2019614072749E+017	[illegible]
87	6.79891637638612E+017	[illegible]
88	1.1000877783661E+018	[illegible]
89	1.77997941600471E+018	[illegible]
90	2.88006719437082E+018	[illegible]
91	4.66004661037553E+018	[illegible]
92	7.54011380474635E+018	[illegible]
93	1.22001604151219E+019	[illegible]
94	1.97402742198682E+019	[illegible]
95	3.19404346349901E+019	[illegible]
96	5.16807088548583E+019	[illegible]
97	8.36211434898484E+019	[illegible]
98	1.35301852344707E+020	[illegible]
99	2.18922995834555E+020	[illegible]
100	3.54224848179262E+020	[illegible]

5. 読み取れること

まず、一見してすぐに気づくような規則性（何かの模様ができるなど）はないので、予想したことを一つずつ確認していく。

①増えていく桁の数

番目	桁	増えた桁
1	1	0
2	1	0
3	2	1
4	2	0
5	3	1
6	4	1
7	4	0
8	5	1
9	6	1
10	6	0
11	7	1
12	8	1
13	8	0
14	9	1
15	10	1
16	10	0
17	11	1
18	12	1
19	13	1
20	13	0
21	14	1
22	15	1
23	15	0
24	16	1
25	17	1
26	17	0
27	18	1
28	19	1
29	19	0
30	20	1
31	21	1
32	22	1
33	22	0
34	23	1
35	24	1
36	24	0
37	25	1
38	26	1
39	26	0
40	27	1
41	28	1
42	28	0
43	29	1
44	30	1
45	31	1
46	31	0
47	32	1
48	33	1
49	33	0
50	34	1
51	35	1
52	35	0
53	36	1
54	37	1
55	38	1
56	38	0
57	39	1
58	40	1
59	40	0
60	41	1
61	42	1
62	42	0
63	43	1
64	44	1
65	44	0
66	45	1
67	46	1
68	47	1
69	47	0
70	48	1
71	49	1
72	49	0
73	50	1
74	51	1
75	51	0
76	52	1
77	53	1
78	53	0
79	54	1
80	55	1
81	56	1
82	56	0
83	57	1
84	58	1
85	58	0
86	59	1
87	60	1
88	60	0
89	61	1
90	62	1
91	63	1
92	63	0
93	64	1
94	65	1
95	65	0
96	66	1
97	67	1
98	67	0
99	68	1
100	69	1

増えていく桁の数は、3,4番ごとに1回、桁の増えないときがあるが100個目までの中では不規則である。

②○と●の数の比

番目	●の数	○の数	割合
1	1	0	100%
2	1	0	100%
3	1	1	50%
4	2	0	100%
5	2	1	67%
6	1	3	25%
7	3	1	75%
8	3	2	60%
9	2	4	33%
10	5	1	83%
11	4	3	57%
12	2	6	25%
13	5	3	63%
14	6	3	67%
15	4	6	40%
16	8	2	80%
17	7	4	64%
18	4	8	33%
19	5	8	38%
20	8	5	62%
21	6	8	43%
22	8	7	53%
23	11	4	73%
24	6	10	38%
25	6	11	35%
26	9	8	53%
27	11	7	61%
28	11	8	58%
29	12	7	63%
30	8	12	40%
31	11	10	52%
32	9	13	41%
33	13	9	59%
34	12	11	52%
35	11	13	46%
36	12	12	50%
37	14	11	56%
38	10	16	38%
39	12	14	46%
40	16	11	59%
41	17	11	61%
42	14	14	50%
43	16	13	55%
44	18	12	60%
45	15	16	48%
46	21	10	68%
47	13	19	41%
48	12	21	36%
49	18	15	55%
50	18	16	53%
51	17	18	49%
52	17	18	49%
53	17	19	47%
54	16	21	43%
55	22	16	58%
56	21	17	55%
57	16	23	41%
58	24	16	60%
59	20	20	50%
60	16	25	39%
61	19	23	45%
62	26	16	62%
63	23	20	53%
64	20	24	45%
65	25	19	57%
66	19	26	42%
67	26	20	57%
68	15	32	32%
69	23	24	49%
70	23	25	48%
71	22	27	45%
72	25	24	51%
73	27	23	54%
74	24	27	47%
75	23	28	45%
76	23	29	44%
77	22	31	42%
78	27	26	51%
79	28	26	52%
80	27	28	49%
81	32	24	57%
82	31	25	55%
83	25	32	44%
84	30	28	52%
85	34	24	59%
86	27	32	46%
87	33	27	55%
88	24	36	40%
89	36	25	59%
90	33	29	53%
91	30	33	48%
92	37	26	59%
93	28	36	44%
94	33	32	51%
95	30	35	46%
96	29	37	44%
97	31	36	46%
98	36	31	54%
99	38	30	56%
100	41	28	59%

素数が多く、比で表すのは難しそうだったので割合で考えることにした。色の意味は
赤・・・100%　黄・・・51～99%　緑・・・50%　青・・・1～49%　水色・・・素数
であり、小数点以下は四捨五入した。色分けしたが、黒と白の現れる割合は残念ながら不規則であった。じゃんけんやサイコロの確率と同じく、数が大きくなる（桁が増える）ほど50%に近くなっていくことはわかった。

③各位の数の合計

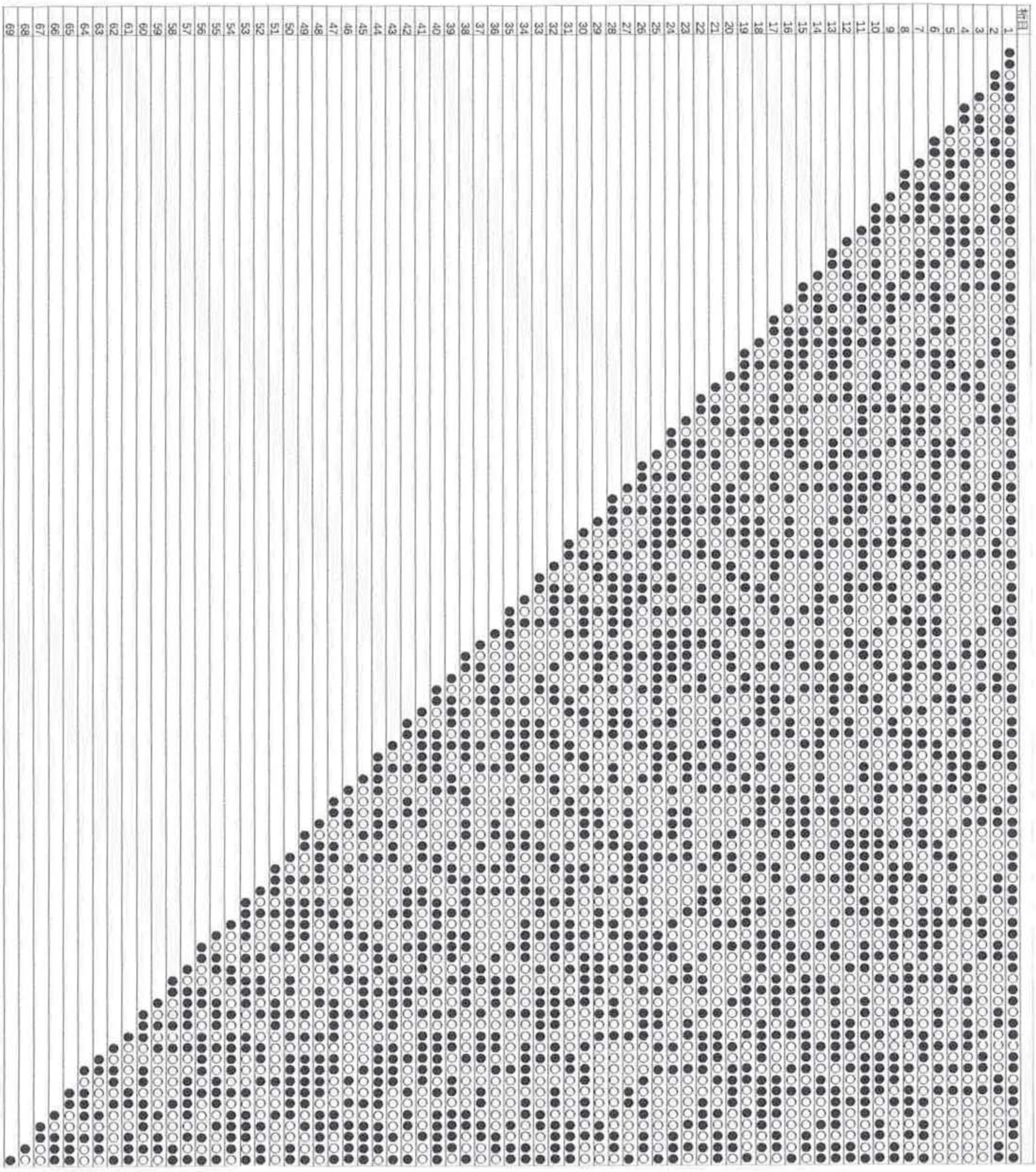
番目	各位の和	増えた数
1	1	0
2	1	0
3	1	0
4	2	1
5	2	0
6	1	-1
7	3	2
8	3	0
9	2	-1
10	5	3
11	4	-1
12	2	-2
13	5	3
14	6	1
15	4	-2
16	8	4
17	7	-1
18	4	-3
19	5	1
20	8	3
21	6	-2
22	8	2
23	11	3
24	6	-5
25	6	0
26	9	3
27	11	2
28	11	0
29	12	1
30	8	-4
31	11	3
32	9	-2
33	13	4
34	12	-1
35	11	-1
36	12	1
37	14	2
38	10	-4
39	12	2
40	16	4
41	17	1
42	14	-3
43	16	2
44	18	2
45	15	-3
46	21	6
47	13	-8
48	12	-1
49	18	6
50	18	0
51	17	-1
52	17	0
53	17	0
54	16	-1
55	22	6
56	21	-1
57	16	-5
58	24	8
59	20	-4
60	16	-4
61	19	3
62	26	7
63	23	-3
64	20	-3
65	25	5
66	19	-6
67	26	7
68	15	-11
69	23	8
70	23	0
71	22	-1
72	25	3
73	27	2
74	24	-3
75	23	-1
76	23	0
77	22	-1
78	27	5
79	28	1
80	27	-1
81	32	5
82	31	-1
83	25	-6
84	30	5
85	34	4
86	27	-7
87	33	6
88	24	-9
89	36	12
90	33	-3
91	30	-3
92	37	7
93	28	-9
94	33	5
95	30	-3
96	29	-1
97	31	2
98	36	5
99	38	2
100	41	3

これは「1」の和なので、●の数の合計と一緒になる。
試しに100までの和を求めてみたが40で、また不規則な結果となった。

④縦の規則性
(表は入り切らなかったため、次ページに掲載)
1～100のフィボナッチ数列を2進数に変換したものを、縦に、つまり下から1桁ずつ見てみた。すると、それぞれの桁に現れる○と●の並び方に規則性があった。

1桁目…●●○
2桁目…●●○○○○
3桁目…●○●●○●○○○○○○
4桁目…●●○○○○●○●●○●●●○●○●○○○○○○○○

このように5桁目以降も桁ごとに1つのパターンが繰り返されていた。
また、そのパターンの長さは3,6,12,24…と1つ桁が上がるごとに倍の長さになっている。そのため、6桁目以降はパターンの方が今回調べたフィボナッチ数列より長く、5桁目までしかパターンが確かめられなかった。


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69

6.考察

今回の研究で見つかった④の規則性について考察する。
まず、何故先程のようなパターンができるのかと、パターンの意味を考えてみる。

★1桁目
パターン…●●○
パターンを2進数にすると…110

2進数で1桁目が「1」になるのは2で割り切れない数のときなので奇数、「0」になるのは2で割り切れる数のときなので偶数である。よって、「●●○」というパターンは、フィボナッチ数列が奇数、奇数、偶数の順番で並んでいることを示している。

★2桁目
パターン…●●○○○○
パターンを2進数にすると…110000

1桁目と同じ考え方だとこれは奇数、奇数、偶数、偶数、偶数、偶数のパターンになるが、これは2桁目なので同じ考え方はできない。10進数で10の位だけで奇数、偶数が判断できないのと同じである。1桁目と合わせて考えると、

○○、●○、○●、●●

の4パターンがあるので、それぞれの意味を考える。

まず、1桁目が「○」である「○○」と「●○」は偶数、「●」である「○●」と「●●」は奇数なことがわかっている。また、「○○」は2回連続で2で割り切れる数であることから、4の倍数、「●○」は1度目は2で割り切れるが2度目は割り切れない数なので4の倍数ではない偶数であることがわかる。4の倍数ではない偶数とは、言い換えると4で割ったときの余りが2の偶数とある。

続いて、「○●」と「●●」の違いだがこれは10進数を2進数に変換する方法を考えると簡単である。「○●」は2で割ったとき、1度目は余りが1出て割り切れないが、2度目は割り切れる数である。その一方で、「●●」は2連続で余りが出る数である。このことから、「○●」は4で割ったときの余りが1、「●●」は3であるという両者の違いがわかる。

よって、この4パターンはそれぞれ

○○…4の倍数。
●○…4で割ったときの余りが2の偶数(2の倍数)。
○●…4で割ったときの余りが1の奇数。
●●…4で割ったときの余りが3の奇数。

を示している。

フィボナッチ数列は最初に「1」が2つありそこには2桁目が存在しないので、2桁目を持つ1桁目のパターンは○から始まる。これを踏まえた上で考えると、1桁目と2桁目を合わせたパターンは

●○ → ●● → ○● → ○○ → ○● → ○●

となる。従って、フィボナッチ数列は

4で割ったときの余りが2の偶数 → 4で割ったときの余りが3の奇数 → 4で割ったときの余りが1の奇数 → 4の倍数 → 4で割ったときの余りが1の奇数 → 4で割ったときの余りが1の奇数

の順番の数列であることがわかる。
また、これは4で割ったときの余りで考えると
2 → 3 → 1 → 0 → 1 → 1
と並んでいることになる。

★3桁目
パターン…●○●●○●○○○○○○
パターンを2進数にすると…101101000000

3桁目だけ単独で考えてはいけないことは2桁目のときと同じなので、1桁目、2桁目と合わせて考える。すると、パターンは

○○○、○●○、●○○、●●○、○○●、○●●、●○●、●●●

となる。それぞれの意味は、2桁目を考えたときの考え方を応用させると、

○○○…8の倍数。
○●○…8で割ったときの余りが2の偶数。
●○○…8で割ったときの余りが4の偶数。
●●○…8で割ったときの余りが6の偶数。
○○●…8で割ったときの余りが1の奇数。
○●●…8で割ったときの余りが3の奇数。
●○●…8で割ったときの余りが5の奇数。
●●●…8で割ったときの余りが7の奇数。

となった。しかし、フィボナッチ数列には「●○○」と「●●○」というパターンは存在せず、

●○● → ○○○ → ●○● → ●○● → ○●○ → ●●● → ○○● → ○○○ → ○○●
→ ○○●→○●○→○●●

という規則となっている。
これは、2桁目を考えたときと同じように8で割った余りに着目すると

5 → 0 → 5 → 5 → 2 → 7 → 1 → 0 → 1 → 1 → 2 → 3

となる。

★4桁目
パターン…●●○○○●○●●○●●●○●○●○○○○○○○
2進数にすると…110001011011101010000000

2, 3桁目から、1桁上がるごとに割る数は2の2乗ずつ大きくなっていくことがわかるので、4桁目は16で割った余りでパターンが決まる。

4桁目のパターンは4桁×24個ありとても長いので、余りのパターンだけ記載すると

8 → 13 → 5 → 2 → 7 → 9 → 0 → 9 → 9 → 2 → 11 → 13 → 8 → 5 → 13 → 2 → 15 → 1 → 0
→ 1 → 1 → 2 → 3 → 5

となる。

★5桁目
パターン…●○●●●○●○●●●●○○○●○●●○●●○●○○○○●●○○○○●○●○●○○○
○○○○○○○
2進数にすると…10111010111100010110110100001100001010100000000000

4桁目のときと同じく、余りのパターンだけ記載する。また、割る数は32である。

13 → 21 → 2 → 23 → 25 → 16 → 9 → 25 → 2 → 27 → 29 → 24 → 21 → 13 → 2 → 15 → 17
→ 0 → 17 → 17 → 2 → 19 → 21 → 8 → 29 → 5 → 2 → 7 → 9 → 2 → 11 → 13 → 24 → 5 → 29
→ 2 → 31 → 1 → 0 → 1 → 1 → 2 → 3 → 5 → 8 → 13

★これらのパターンからわかること

1～5桁目のパターンを見て、

・1桁目より、3番ごとに2の倍数（偶数）
・2桁目より、6番ごとに4の倍数
・3桁目より、6番ごとに8の倍数
・4桁目より、12番ごとに16の倍数
・5桁目より、24番ごとに32の倍数

がフィボナッチ数列には登場することがわかる。また、2, 3桁目の結果より4の倍数だが8の倍数でない数、例えば12などはフィボナッチ数列に出てこないこともわかる。そして、これらの規則から

・48番ごとに64の倍数
・96番ごとに128の倍数

などが現れることが予測できる。

7.まとめ

フィボナッチ数列では、
・3番ごとに偶数が現れる。
・6番ごとに8の倍数が現れる。
・$n>1$のとき、3×2^n番ごとに4×2^nの倍数が現れる。

8.感想

私はパソコンがかなり好きだ。だからフィボナッチ数列を2進数に変換して調べてみるなんてことを思いついたのだと思う。思いついた当初、この研究をやりたくて仕方がなくなったが、その一方で何も法則が見つからなかったらどうしよう、という不安もあった。しかし、実際に調べてみると0と1の並び方から規則を見つけ出すことができた。今回見つかった法則は、もしかしたら数学的に証明できてしまうようなものかもしれないが、それでも2進数に変換するという方法を使ってこの規則を見つけられたことは達成感があり、嬉しかった。

フィボナッチ数列は数が大きくなるのが早く、79番目にして兆の位を超えてしまう。そんな数列から法則を見つけ出すのは、容易ではないと思う。今回私が取った方法のように、2進数に変換したり、またコンピュータを使ったりと様々な方向からアプローチすることでそれまで分からなかったことが見えてくるのだろうと感じた。

2進数の場合、1つ位が上がるごとに数は2倍になっていくが、他の進数だとまた別の増え方をする。今回は2の累乗がパターンとして出てきたが、他の進数ではその進数に合った、今回とはまた違った規則が見つかりそうだと思った。

自然界に多く存在する不思議な数列であるフィボナッチ数列と、コンピュータが効率よく動くために最適な数である2進数。このような真反対である2つを合わせて考えても規則が見えてくるところが面白いと思った。これからも今回のように、自分の不思議に思ったことや興味を持ったことを自分なりに研究していきたい。

*[1]…下記のプログラムを自作した。使用したプログラミング言語は Python3。
*[2]…「高校数学+α　基礎と論理の物語」　宮腰忠 著　共立出版
　のＰ32～Ｐ34を参考にした。
*[3]…表では79番目以降が表作成ソフトの表示可能桁数を超えたため指数表記になっているが、2進数に変換する際に使用した数は通常の数値であった。

```
i = 0  # カウンタ
a = 0
b = 1
fib_ls = list()  # 2進数のフィボナッチ数のリスト

with open("binary.txt", "w") as fout:  # 書き込み用ファイル
  while(i < 100):
    a, b = b, a+b
    fib = a  # フィボナッチ数
    ls = str()  # 余りを入れるリスト

    while(fib > 0):  # 2進数を求める
      remainder = str(fib % 2)  # 2で割った余り
      ls += remainder  # 余りをリストに追加
      fib = fib // 2  #2 で割る

    binary = ls[::-1]  # 余りの順番を逆に

    fib_ls.append(binary)  # フィボナッチ数のリストに追加
    i += 1  # カウンタ

  for line in fib_ls:  # フィボナッチ数を一つずつ処理
    fib = [x for x in list(line)]  # 一文字ずつに分ける
    for x in fib:  # 一文字ずつ処理
      if x == "0":
        fout.write("○")  #0 なら○を書き込む
      elif x == "1":
        fout.write("●")  #1 なら●を書き込む

    fout.write("\n")
```