УДК 511.75:517.13

КОНЦЕПЦИЯ СЧАСТЛИВЫХ ЧИСЕЛ КАК ОБЪЕКТ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Аврусевич В.В, Мазалевич П.В., студенты гр.451003

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баркова Е.А. – канд. физ.-мат. наук, доцент

Аннотация. В работе исследуются счастливые числа – натуральные числа, сумма квадратов цифр которых в итоге даёт 1. Разработан алгоритм их поиска и программное средство для анализа больших диапазонов (до 289 млн за 86 мин). Предложен метод шифрования данных на основе матриц из счастливых чисел, включающий кодирование, умножение на матрицу и обратное преобразование. Показана эффективность метода на примерах. Результаты могут применяться в криптографии и генерации псевдослучайных последовательностей.

Ключевые слова. Счастливые числа, несчастливые числа, алгоритм, шифрование данных, таблица ASCII, матричное преобразование, генерация случайных чисел.

Что такое счастливое число? Может быть, цифра 7? Исходя из поверий – да, однако в своей научной работе мы рассмотрим этот вопрос с другой стороны.

В математике счастливое число – это любое положительное целое число, сумма квадратов цифр которого в конечном счёте приводит к 1. Но если есть счастливые числа, то вероятно будут и несчастливые. Несчастливое число – это также любое положительное целое число, за исключением того, что при сложении квадратов его цифр, 1 никогда не получится.

Эта тема заинтересовала нас, так как такие числа могут использоваться для генерации случайных чисел и шифрования данных, что широко используется в IT-сфере.

Сперва определим, как же всё-таки находить такие числа. Рассмотрим ранее упоминавшееся число 7. Обозначим за S(7) сумму квадратов цифр выбранного числа.

На первом шаге получим, что

$S(7) = 7^2 = 49$

Повторим действия:

$S(49) = 4^2 + 9^2 = 16 + 81 = 97$

$S(97) = 9^2 + 7^2 = 81 + 49 = 130$

$S(130) = 1^2 + 3^2 + 0^2 = 1 + 9 + 0 = 10$

$S(10) = 1^2 + 0^2 = 1 + 0 = 1$

На 5 шаге сумма стала равна 1, что и подтверждает то, что число 7 – счастливое. Что насчёт следующего счастливого числа?

Проверим тоже самое для 8:

$S(8) = 8^2 = 64$

$S(64) = 6^2 + 4^2 = 36 + 16 = 52$

$S(52) = 5^2 + 2^2 = 25 + 4 = 29$

$S(29) = 2^2 + 9^2 = 4 + 81 = 85$

$S(85) = 8^2 + 5^2 = 64 + 25 = 89$

$S(89) = 8^2 + 9^2 = 64 + 81 = 145$

$S(145) = 1^2 + 4^2 + 5^2 = 1 + 16 + 25 = 42$

$S(42) = 4^2 + 2^2 = 16 + 4 = 20$

$S(20) = 2^2 + 0^2 = 4 + 0 = 4$

$S(4) = 4^2 = 16$

$S(16) = 1^2 + 6^2 = 1 + 36 = 37$

$S(37) = 3^2 + 7^2 = 9 + 49 = 58$

$S(58) = 5^2 + 8^2 = 25 + 64 = 89$

На 13 шаге получаем тот же результат, что и на 5. Логично, что если снова захотим просуммировать суммы квадратов цифр числа 89, то повторим все вычисления, которые выполняли на 6 и следующих после него шагах. Это и есть цикл, который никогда не приведёт к 1. Следовательно, число 8 – несчастливое.

Определим алгоритм для нахождения счастливых чисел на основе примера:

Пусть n – любое положительное целое число. Тогда, чтобы определить “счастливость” этого числа, возьмём его цифры и просуммируем их квадраты:

$$n = m_1, m_2, \ldots, m_k, \quad (1)$$

где m_i – цифра числа n, k – количество цифр числа n.

$$S(n) = m_1^2 + m_2^2 + \ldots + m_k^2 \quad (2)$$

Если S(n) = 1, тогда число счастливое.
Если S(n) ≠ 1 и уже встречалось на предыдущих шагах, то число несчастливое.
Если S(n) ≠ 1 и ещё не встречалось, то посчитать S(S(n))

В процессе вычислений можно обнаружить некоторые свойства счастливых/несчастливых чисел. Например, заметно, что если проверяемое число счастливое, то числа, полученные при вычислениях суммы цифр на отдельных шагах, также являются счастливыми. Аналогично и с несчастливыми числами: числа, полученные на промежуточных этапах, являются несчастливыми. Можно заметить, что на “счастливость” числа не влияет перестановка его цифр и наличие нулей.

Для того, чтобы упростить поиск счастливых чисел, мы разработали программное средство. Пользователь должен ввести в поле нужный диапазон и прожать кнопку для получения результата. Для расчёта времени поиска мы ввели диапазон от 300 до 290 000 000 и засекли время. Все счастливые числа были найдены спустя 86 минут. Интерфейс программного средства представлен на рисунке 1.

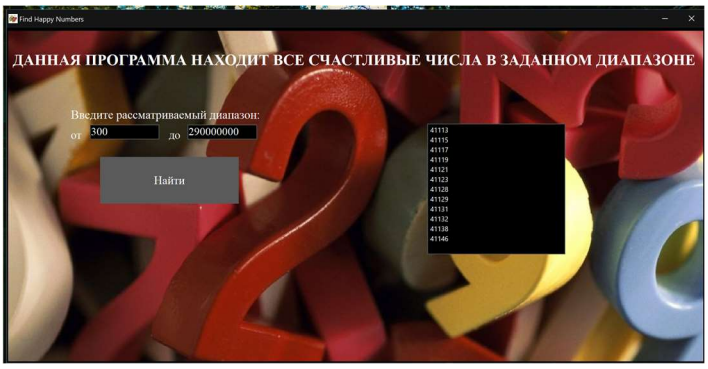

Рисунок 1 – Общий вид интерфейса программного средства

При поиске были выявлены зависимости, представленные на рисунках 2 и 3.

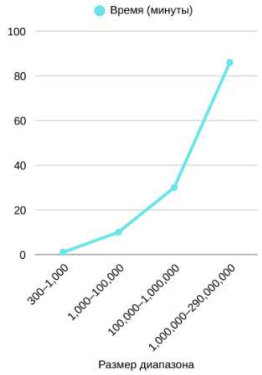

Рисунок 2 – График зависимости диапазона от времени, затрачиваемого на поиск таких чисел

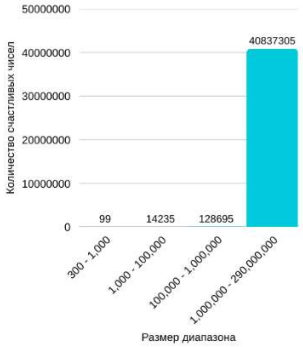

Рисунок 3 – График зависимости диапазона от количества счастливых чисел на нём

Рассмотрим на примере, как можно применить счастливые числа для шифрования данных.

Допустим, мы хотим зашифровать слово “MEOW”. Для этого преобразуем буквы слова в соответствующие коды по таблице ASCII, представленной на рисунке 4.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Рисунок 4 – Таблица ASCII

M – 77, E – 69, O – 79, W – 87. Обозначим полученные коды за элементы вектора-строки: V = [77, 69, 79, 87]. Выпишем квадратную матрицу из счастливых чисел, размера n, где n – длина слова, а именно: M = $\begin{pmatrix} 1 & 3 & 7 & 10 \\ 3 & 7 & 10 & 13 \\ 7 & 10 & 13 & 19 \\ 10 & 13 & 19 & 23 \end{pmatrix}$. Так как размер вектора-строки 1 × 4 и размер матрицы 4 × 4 (количество столбцов вектора совпадает с количеством строк в матрице), перемножим вектор на матрицу и получим новый вектор C:

$$C = V \times M = [77, 69, 79, 87] \times \begin{pmatrix} 1 & 3 & 7 & 10 \\ 3 & 7 & 10 & 13 \\ 7 & 10 & 13 & 19 \\ 10 & 13 & 19 & 23 \end{pmatrix} = [1707, 2635, 3909, 5169] \quad (3)$$

Разобьём каждый полученный элемент по цифрам: 1707 – 1, 7, 0 и 7; 2635 – 2, 6, 3 и 5; 3909 – 3, 9, 0 и 9; 5169 – 5, 1, 6 и 9. Если бы вектор содержал элементы с нечётным количеством цифр, мы бы приписали 0 спереди. К каждому числу прибавим 33 (т.к. 1 – 32 символы таблицы ASCII непечатные), сочтём их за коды и преобразуем в символы по этой же таблице:

1 + 33 = 34 (код символа “”)
7 + 33 = 40 (код символа ‘(’)
0 + 33 = 33 (код символа ‘!’)
Для 7 уже считали, получился символ ‘(’
2 + 33 = 35 (код символа ‘#’)
6 + 33 = 39 (код символа “’)
3 + 33 = 36 (код символа ‘$’)
5 + 33 = 38 (код символа ‘&’)
Для 3 уже считали, получился символ ‘$’
9 + 33 = 42 (код символа ‘*’)
Для 0 уже считали, получился символ ‘!’
Для 9 уже считали, получился символ ‘*’
Для 5 уже считали, получился символ ‘&’
Для 1 уже считали, получился символ “”
Для 6 уже считали, получился символ “’
Для 9 уже считали, получился символ ‘*’

Таким образом, мы зашифровали слово “MEOW” в ”(!(#’$&$*!*&”’*. Как теперь расшифровать, например, такой набор символов: “&*!#&#’$(&%&!””? Найдём код каждого символа по вышеупомянутой таблице: “ – 34, & – 38, * – 42, ! – 33, # – 35, ‘ – 39, $ – 36, (– 40, % – 37. Отнимем от каждого числа 33 :

34 – 33 = 1
38 – 33 = 5
42 – 33 = 9
33 – 33 = 0

35 – 33 = 2
39 – 33 = 6
36 – 33 = 3
40 – 33 = 7
37 – 33 = 4

Тогда получим 1590252637545011. Логично, что число символов в уже зашифрованной последовательности всегда будет чётным, поэтому разобьём результат по 4 цифры:
1590 2526 3754 5011. Пусть эти числа будут элементами вектора V’ = [1590, 2526, 3754, 5011]. Тогда для того, чтобы расшифровать последовательность, найдём обратную матрицу к матрице M и умножим вектор на неё (т.к. число элементов вектора равно размерности матрицы M из счастливых чисел и матрица M – квадратная, то можем взять её для примера).

Найдём определитель матрицы M:

$\det M = \begin{vmatrix} 1 & 3 & 7 & 10 \\ 3 & 7 & 10 & 13 \\ 7 & 10 & 13 & 19 \\ 10 & 13 & 19 & 23 \end{vmatrix}$. С помощью элементарных преобразований приводим матрицу к верхнетреугольному виду и с лёгкостью находим её определитель:

$\begin{vmatrix} 1 & 3 & 7 & 10 \\ 0 & -2 & -11 & 17 \\ 0 & 0 & 24,5 & 42,5 \\ 0 & 0 & 0 & -6\frac{11}{49} \end{vmatrix}$ = 1 * (-2) * 24,5 * (-$6\frac{11}{49}$) = 305 ≠ 0. Определитель не равен нулю,

следовательно, обратная матрица существует. Для вычисления обратной матрицы припишем к исходной справа единичную:. Применим элементарные преобразования над строками, чтобы привести левую часть матрицы к единичной: $\left(\begin{array}{cccc|cccc} 1 & 3 & 7 & 10 & 1 & 0 & 0 & 0 \\ 3 & 7 & 10 & 13 & 0 & 1 & 0 & 0 \\ 7 & 10 & 13 & 19 & 0 & 0 & 1 & 0 \\ 10 & 13 & 19 & 23 & 0 & 0 & 0 & 1 \end{array}\right)$. Правая часть получившейся матрицы и есть обратная к M матрице матрица M’. Перемножим вектор и обратную матрицу:

$$V = V' \times M' = [1590, 2526, 3754, 5011] \times \begin{pmatrix} \frac{9}{305} & -\frac{143}{305} & \frac{7}{61} & \frac{48}{305} \\ -\frac{143}{305} & \frac{171}{305} & \frac{4}{61} & -\frac{51}{305} \\ \frac{7}{61} & \frac{4}{61} & -\frac{27}{61} & \frac{17}{61} \\ \frac{48}{305} & -\frac{51}{305} & \frac{17}{61} & -\frac{49}{305} \end{pmatrix} = [82, 79, 83, 69] \quad (4)$$

По таблице ASCII: 82 – R, 79 – O, 83 – S, 69 – E.

В итоге последовательность символов “&*!#&#’$(&%&!”” оказалась словом “ROSE”. Наш метод наглядно показывает, что данные можно шифровать с помощью счастливых чисел.

В ходе выполненного исследования была тщательно изучена концепция счастливых чисел как объекта математического анализа и их свойства. Разработанный алгоритм стал основой для реализации программного средства. Это средство позволило автоматизировать процесс поиска счастливых чисел в широких числовых диапазонах. Также был создан метод шифрования данных с помощью счастливых чисел. Он доказал свою эффективность при попытке расшифровать данные.

Список использованных источников:

1. Guy, R. K. Happy Numbers: A Mathematical Exploration / R. K. Guy // Unsolved Problems in Number Theory / ed. R. K. Guy. – 3rd ed. – New York: Springer, 2004. – P. 103 – 114.

2. Grundman, H. G. Iterated Sums of Squares of Digits / H. G. Grundman, E. A. Teeple // Fibonacci Quarterly. – 2002. – Vol. 3. – P. 251 – 256.

UDC 511.75:517.13

THE CONCEPT OF LUCKY NUMBERS AS AN OBJECT OF MATHEMATICAL ANALYSIS

Aurusevich V.V., Mazalevich P.V

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Barkova E.A. – PhD in Physics and Mathematics, Associate Professor

Annotation. The study investigates happy numbers - natural numbers whose sum of squared digits eventually equals 1. An algorithm for their detection was developed along with software capable of analyzing large ranges (up to 290 million in 86 minutes). A data encryption method based on matrices of happy numbers was proposed, involving encoding, matrix multiplication, and inverse transformation. The method's effectiveness was demonstrated through examples. The results may find applications in cryptography and pseudorandom number generation.

Keywords. Happy numbers, unhappy numbers, algorithm, data encryption, ASCII table, matrix transformation, random number generation.