Pensées 7

Ashley E. Vaughn

February 13, 2024

Contents

1 Concept **3**

1.1 What are Repetends? 3

1.2 Selected Bases 3

2 Methods **4**

2.1 Code 4

3 Data **5**

3.1 2^3 5

3.2 2^4 5

3.3 2^5 5

3.4 2^6 5

3.5 2^7 5

3.6 2^8 6

3.7 2^9 6

3.8 2^{10} 6

3.9 2^{11} 6

3.10 2^{12} 6

3.11 2^3 Digits 7

3.12 2^{12} Digits 7

3.13 2^{12} Top 25 Bases 2 - 4096 7

4 Conclusion **8**

1 Concept

1.1 What are Repetends?

The reciprocals of integers with prime factors which the base lacks have an infinite number of digits after the decimal point which repeat in units called repetends. The length of the repetend is the number of digits which constitute the unit which repeats, which generally decreases with base size, but there are disproportionate effects from the properties of the base itself. A base would ideally minimize the length of these repetends to make inference easier.

1/2	0.8	2	1/2
1/3	$0.\overline{3}$	$0.\overline{5}$	1/3
1/4	0.25	0.4	1/4
1/5	0.2	$0.\overline{3}$	1/5
1/6	$0.1\overline{6}$	$0.2\overline{A}$	1/6
1/7	$0.\overline{142857}$	$0.\overline{249}$	1/7
1/8	0.125	0.2	1/8

1.2 Selected Bases

2	3	4	6	8	9	10	12	16	21	25	27	29
30	32	34	36	45	46	49	55	56	57	60	64	

These bases are sampled based on popularity and retroactive selection based on performance in metrics.

2 Methods

All the metrics are based on the mean number of digits in the repetends of $1/n$ in base b for $2 \leq n \leq 2^m$.

2.1 Code

```
import numpy as np
from numba import njit


@np.vectorize
@njit
def repetend_digits(n: int, base: int):
    remainder_set = set()
    remainder = 1 % n
    repetend_length = 0

    while remainder != 0 and remainder not in remainder_set:
        remainder_set.add(remainder)
        remainder = (remainder * base) % n
        repetend_length += 1

    return repetend_length


bases = np.array([2,3,4,6,8,9,10,12,16,21,25,27,29,30,32,
                  34,36,45,46,49,55,56,57,60,64])

# n = np.expand_dims(np.array(range(2, 2**n + 1)), axis=-1)
# t = repetend_digits(n, bases).T
# bases[(bases ** t.mean(axis=1)).argsort()]
```

3 Data

Displayed are the bases sorted by bases to the power of mean repetend digits, lower is better. This approximately corresponds to the numerical size of the repetend and is the most meaningful measure, also accounting for the advantage larger bases have.
Radix economy is the number of digits in a number times the base:

$$E(b, N) = b\lfloor \log_b(N) + 1 \rfloor$$

The number of digits is approximately:

$$n \approx \log_b(N) = \ln(N)/\ln(b)$$

Thus the metric is approximately the number:

$$N \approx b^n$$

3.1 2^3

2	4	6	3	9	8	21	16	36	25
49	10	29	64	12	55	56	30	60	57
34	27	46	32	45					

3.2 2^4

2	4	3	9	21	16	6	25	8	10
55	12	36	64	49	60	34	27	56	29
30	45	57	32	46					

3.3 2^5

2	4	3	6	9	16	25	8	36	30
12	64	10	49	21	45	55	57	27	29
46	60	34	56	32					

3.4 2^6

2	4	3	9	16	6	8	25	36	64
49	10	12	21	27	30	46	57	29	45
60	34	32	55	56					

3.5 2^7

2	4	3	9	16	8	6	25	64	10
36	49	12	21	27	30	32	45	34	57
46	55	29	60	56					

3.6 2^8

2	4	3	16	9	8	6	64	25	10
36	12	49	27	21	30	32	34	60	46
55	45	29	57	56					

3.7 2^9

2	4	3	16	9	8	6	64	25	36
10	49	12	27	21	30	32	34	55	45
29	60	46	56	57					

3.8 2^{10}

2	4	3	16	9	8	6	64	25	36
10	49	12	27	21	32	30	34	29	55
46	45	57	60	56					

3.9 2^{11}

2	4	3	16	9	8	6	64	25	36
10	49	12	27	21	32	30	34	29	45
55	46	57	60	56					

3.10 2^{12}

2	4	3	16	9	8	64	6	25	12
57	56	55	49	46	45	34	10	32	30
60	27	21	36	29					

3.11 2^3 Digits

64	16	36	49	25	4	9	8	27	32
21	30	46	55	34	57	45	60	10	56
6	12	29	2	3					

3.12 2^{12} Digits

64	16	36	49	25	4	9	8	27	32
21	55	57	30	34	45	46	60	56	10
6	12	29	3	2					

3.13 2^{12} Top 25 Bases 2 - 4096

4096	64	729	2401	1296	256	16	1024	81	625
441	3025	3249	900	1521	2500	3136	1156	2025	1936
676	225	2601	2116	324					

The first base to beat 64 in terms of number of digits is 4096, showing that it is an extremely good base in terms of raw numbers of digits in repetends. In addition, 16 holds a very good position, being the smallest base in the top 25.

4 Conclusion

2, 4, and 16 have an advantage in terms of repetend sizes which increases with the size of n but holds for all reasonable ranges. Binary and quaternary have the disadvantage of a large number of digits in their repetends even though they are numerically small, but hexadecimal has small repetends both in size and length, and thus is easier to remember and use for inference. Binary would require some modifications to existing systems for human use, while hexadecimal is much closer to our current system, decimal. Thus, binary, quaternary, and hexadecimal are the best systems in terms of repetends, contrary to the perception that bases with more prime factors are superior.