Pensées 7

Ashley V.

Concept

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 of a fraction from its repetend easier. Example: $0.\overline{3}\ldots$ can easily be inferred to be $\frac{1}{3}$, whereas $0.\overline{0344827586206896551724137931}\ldots$ is more difficult.

Square Numbers

We have found that almost all competitive bases are square numbers, numbers n such that $n = k^2, k \in \mathbb{N}$. The selected bases follow:
$[2, 4, 9, 10, 12, 16, 25, 36, 49, 64]$

Methods

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

Code

```
use gcd::Gcd;

/// prime factors of n
pub fn prime_facts(mut n: u128) -> Vec<u128> {
    let mut facts = Vec::new();

    let mut i = 2;
    while i <= n {
        while n % i == 0 {
            facts.push(i);

            n /= i;
        }

        i += 1;
    }

    facts
}
```

```rust
/// smallest k such that a^k == 1 (mod n)
pub fn mult_order(a: u128, n: u128) -> Result<u128, String> {
    // a^k % 1 = 0 because a in Z
    assert!(n >= 2);

    assert!(a.gcd(n) == 1);

    let a_mod_n = a % &n;

    let mut r = 1;
    for k in 1..n {
        r = r * &a_mod_n % &n;

        if r == 1 {
            return Ok(k);
        }
    }

    Err("could not get order".to_string())
}

/// period of decimal representation of 1/n, zero if terminating
pub fn repetend_len(mut n: u128, base: u128) -> Result<u128, String> {
    let mut base_facts = prime_facts(base);
    base_facts.dedup();

    for b in base_facts {
        while n % b == 0 {
            n /= b;
        }
    }

    if n == 1 {
        return Ok(0);
    }

    mult_order(base, n)
}
```

```rust
use rayon::prelude::*;
use repetend_len::repetend_len;

fn main() {
    let bases = vec![4, 9, 10, 12, 16, 25, 36, 49, 64, 256, 4096];
    let n = 1 << 16;

    let mut lens = bases
        .par_iter()
        .map(|base| {
            println!("running: {base}");
            (2..n)
                .map(|i| repetend_len(i, *base).unwrap())
                .sum::<u128>() as f64
                / (n - 2) as f64
        })
        .zip(bases.par_iter())
        .collect::<Vec<_>>();
    lens.sort_by(|a, b| a.partial_cmp(b).unwrap());

    let mut sizes = lens
        .iter()
        .map(|(l, base)| (l * (**base as f64).log2(), base))
        .collect::<Vec<_>>();
    sizes.sort_by(|a, b| a.partial_cmp(b).unwrap());

    println!("lens:\n{:#?}", lens);
    println!("sizes:\n{:#?}", sizes);
}
```

Metrics

The “length” metric is the mean repetend length. The “size” metric is monotonic to $\text{base}^{\text{length}}$, related to the numerical size of the repeatand, but, because log is monotonic, it can be sorted by $\text{length} \log_2(\text{base})$. It’s necessary to take into consideration that this is logarithmic when comparing results.

Results

The first row is the base, the second is the value, and the third is relative to the middle value.

2..2^3 Len

36	64	16	49	4	25	9	10	12	2
0.33	0.83	1	1	1.17	1.17	1.33	1.33	1.67	1.83
0.29	0.71	0.86	0.86	1	1	1.14	1.14	1.43	1.57

2..2^3 Size

36	2	4	16	9	10	64	25	49	12
1.72	1.83	2.33	4	4.23	4.43	5	5.42	5.61	5.97
0.41	0.43	0.55	0.95	1	1.05	1.18	1.28	1.33	1.41

2..2^4 Len

36	64	16	25	9	10	12	49	4	2
1.14	1.36	1.64	1.71	1.79	1.79	1.93	1.93	2.07	3.71
0.64	0.76	0.92	0.96	1	1	1.08	1.08	1.16	2.08

2..2^4 Size

2	4	9	36	10	16	12	25	64	49
3.71	4.14	5.66	5.91	5.93	6.57	6.91	7.96	8.14	10.83
0.63	0.7	0.95	1	1	1.11	1.17	1.34	1.37	1.83

2..2^5 Len

36	64	16	25	49	9	4	12	10	2
2.47	2.8	3.03	3.07	3.47	3.8	3.87	4.57	5	6.9
0.71	0.81	0.87	0.88	1	1.1	1.12	1.32	1.44	1.99

2..2^5 Size

2	4	9	16	36	25	12	10	64	49
6.9	7.73	12.05	12.13	12.75	14.24	16.37	16.61	16.8	19.46
0.54	0.61	0.94	0.95	1	1.12	1.28	1.3	1.32	1.53

2..2^6 Len

36	64	16	49	25	9	4	12	10	2
4.76	4.77	5.06	5.68	5.89	6.31	6.71	8.87	9.35	12
0.81	0.81	0.86	0.96	1	1.07	1.14	1.51	1.59	2.04

2..2^6 Size

2	4	9	16	36	25	64	10	12	49
12	13.42	19.99	20.26	24.6	27.34	28.65	31.08	31.8	31.88
0.49	0.55	0.81	0.82	1	1.11	1.16	1.26	1.29	1.3

2..2^7 Len

64	16	36	25	49	9	4	10	12	2
7.94	8.63	9.58	10.43	10.69	10.88	11.2	14.87	16.78	19.99
0.74	0.81	0.9	0.98	1	1.02	1.05	1.39	1.57	1.87

2..2^7 Size

2	4	9	16	64	25	10	36	49	12
19.99	22.4	34.49	34.54	47.67	48.43	49.41	49.52	60.02	60.15
0.42	0.47	0.72	0.72	1	1.02	1.04	1.04	1.26	1.26

2..2^8 Len

64	16	36	25	49	4	9	10	12	2
13.67	15.51	17.09	18.37	19.06	19.67	20.09	26.61	28.69	35.24
0.72	0.81	0.9	0.96	1	1.03	1.05	1.4	1.51	1.85

2..2^8 Size

2	4	16	9	64	25	36	10	12	49
35.24	39.35	62.05	63.67	82.04	85.31	88.36	88.41	102.85	106.99
0.43	0.48	0.76	0.78	1	1.04	1.08	1.08	1.25	1.3

2..2^9 Len

64	16	36	49	25	4	9	12	10	2
24.61	27.58	30.21	33.04	33.97	35.44	36.33	52.64	52.94	62.67
0.72	0.81	0.89	0.97	1	1.04	1.07	1.55	1.56	1.84

2..2^9 Size

2	4	16	9	64	36	25	10	49	12
62.67	70.88	110.34	115.17	147.64	156.2	157.75	175.85	185.48	188.72
0.42	0.48	0.75	0.78	1	1.06	1.07	1.19	1.26	1.28

2..2^10 Len

64	16	36	49	25	4	9	10	12	2
43.29	48.5	57.11	60.79	62.76	63.54	65.78	97.41	99.05	113.63
0.69	0.77	0.91	0.97	1	1.01	1.05	1.55	1.58	1.81

2..2^10 Size

2	4	16	9	64	25	36	10	49	12
113.63	127.08	193.99	208.51	259.76	291.44	295.26	323.57	341.33	355.08
0.44	0.49	0.75	0.8	1	1.12	1.14	1.25	1.31	1.37

2..2^11 Len

64	16	36	49	25	4	9	10	12	2
78.54	89.3	105.49	112	114.42	116.54	119.92	179.38	187.19	208.15
0.69	0.78	0.92	0.98	1	1.02	1.05	1.57	1.64	1.82

2..2^11 Size

2	4	16	9	64	25	36	10	49	12
208.15	233.08	357.2	380.14	471.23	531.35	545.38	595.89	628.85	671.06
0.44	0.49	0.76	0.81	1	1.13	1.16	1.26	1.33	1.42

2..2^12 Len

64	16	36	49	25	4	9	10	12	2
145.4	165.29	198.65	205.07	216.11	217.09	221.02	339.62	345.96	390.79
0.67	0.76	0.92	0.95	1	1	1.02	1.57	1.6	1.81

2..2^12 Size

2	4	16	9	64	25	36	10	49	12
390.79	434.18	661.17	700.62	872.37	1003.58	1027	1128.2	1151.43	1240.27
0.45	0.5	0.76	0.8	1	1.15	1.18	1.29	1.32	1.42

2..2^13 Len

64	16	36	49	25	4	9	10	12	2
269.22	305.19	371.45	391.74	397.15	401.52	411.48	634.95	653.2	723.72
0.68	0.77	0.94	0.99	1	1.01	1.04	1.6	1.64	1.82

2..2^13 Size

2	4	16	9	64	25	36	10	49	12
723.72	803.03	1220.75	1304.35	1615.3	1844.3	1920.38	2109.25	2199.49	2341.68
0.45	0.5	0.76	0.81	1	1.14	1.19	1.31	1.36	1.45

2..2^14 Len

64	16	36	49	25	4	9	10	12	2
500.3	566.72	701.35	736.8	739.63	745.94	774.09	1199.54	1238.45	1344.47
0.68	0.77	0.95	1	1	1.01	1.05	1.62	1.67	1.82

2..2^14 Size

2	4	16	9	64	25	36	10	49	12
1344.47	1491.88	2266.89	2453.8	3001.8	3434.73	3625.9	3984.78	4136.94	4439.79
0.45	0.5	0.76	0.82	1	1.14	1.21	1.33	1.38	1.48

2..2^15 Len

64	16	36	49	25	4	9	10	12	2
942.21	1062.25	1323.53	1384.23	1391.61	1405.18	1459.45	2259.42	2363.43	2539.66
0.68	0.76	0.95	0.99	1	1.01	1.05	1.62	1.7	1.82

2..2^15 Size

2	4	16	9	64	25	36	10	49	12
2539.66	2810.37	4249	4626.34	5653.24	6462.42	6842.55	7505.63	7772.06	8472.82
0.45	0.5	0.75	0.82	1	1.14	1.21	1.33	1.37	1.5

2..2^16 Len

64	16	36	49	25	4	9	10	12	2
1776.15	2008.32	2499.92	2617.77	2650.27	2660.5	2774.23	4302.73	4467.83	4803.49
0.67	0.76	0.94	0.99	1	1	1.05	1.62	1.69	1.81

2..2^16 Size

2	4	16	9	64	25	36	10	49	12
4803.49	5321	8033.28	8794.1	10656.92	12307.47	12924.42	14293.36	14698.02	16016.99
0.45	0.5	0.75	0.83	1	1.15	1.21	1.34	1.38	1.5

2..2^17 Len

64	16	36	49	4	25	9	10	12	2
3368.04	3812.41	4756.72	4973.38	5050.96	5051.38	5284.08	8194.04	8530.87	9127.63
0.67	0.75	0.94	0.98	1	1	1.05	1.62	1.69	1.81

2..2^17 Size

2	4	16	9	64	25	36	10	49	12
9127.63	10101.91	15249.64	16750.14	20208.23	23457.89	24591.9	27220.01	27924.11	30582.85
0.45	0.5	0.75	0.83	1	1.16	1.22	1.35	1.38	1.51

2..2^18 Len

64	16	36	49	25	4	9	10	12	2
6408.36	7253.84	9096.14	9496.45	9624.37	9627.57	10085.04	15699.18	16343.94	17410.73
0.67	0.75	0.95	0.99	1	1	1.05	1.63	1.7	1.81

2..2^18 Size

2	4	16	9	64	25	36	10	49	12
17410.73	19255.14	29015.36	31968.83	38450.18	44694.2	47026.34	52151.56	53319.81	58592.42
0.45	0.5	0.75	0.83	1	1.16	1.22	1.36	1.39	1.52

Conclusion

The scaling of repetend lengths and sizes in various bases generally favors bases which are mid-sized powers of 2 like 16 or 64, but 4 is also a viable option, especially in terms of size, where it's nearly the same as 2, while also having a competitive total length. Overall, the winner in terms of these metrics and also usage considerations is 16 (hexadecimal). At lower values of n the dominant factor is likely the prime factors of the base resulting in termination, but at higher values those that perform the best only have 2 as a prime factor, suggesting other influences which are more important for larger values. Repetends are another factor of fractions in various bases that should be taken into account in weighing the pros and cons of different bases, beyond just the number of prime factors.