

digits of powers of 2

Ben Bolker

21:19 03 January 2016

Random thought #1, for an interesting math programming problem.
What is the distribution of digits of powers of 2?

Are they uniform, or do they have some interesting pattern? At the beginning (2,4,8,16) the sequence is odd-free; the first “3” occurs at $2^5=32$, the first “5” at $2^8=256$, the first “7” at 2^{15} (but the first “9” at 2^{12}). Does the rarity of odd digits continue?

In its simplest form, this is a very easy program to write: you just need to loop over powers of 2 and use string processing to accumulate digits.

The patterns are probably more interesting if we keep track of them by digit. For example, it’s trivial to figure out the least-significant digit patterns, and not too hard to figure out the most-significant digit patterns:

```
res = []    ## empty list
for x in range(20):
    res += format(2**x)[-1]
print(res)

## ['1', '2', '4', '8', '6', '2', '4', '8', '6', '2', '4', '8', '6', '2', '4', '8', '6', '2', '4', '8']
```

You can also do this slightly more compactly with a more-advanced Python feature called a list comprehensions:

```
[format(2**x)[-1] for x in range(20)]
```

The most significant digits also have a pattern, although it’s a little bit less obvious:

```
print([format(2**x)[0] for x in range(20)])

## ['1', '2', '4', '8', '1', '3', '6', '1', '2', '5', '1', '2', '4', '8', '1', '3', '6', '1', '2', '5']
```

Next things to do/extensions:

- store the results as a list averaged by number of digits, so that we can go through and pick out the first/last/second-to-last etc. digit.
- draw pictures
- think about *computational complexity* of this problem, and do some timing to test the conjecture that it’s $O(n \log n)$