Smallest Good Base

Matei-Alexandru Gardus

September 27, 2020

Contents

1 Problem

1.1 Directions

Problem Link

> For an integer n, we call $k \geq 2$ a good base of n, if all digits of n base k are 1.
>
> Now given a string representing n, you should return the smallest good base of n in string format.

2 Notes

This is pure math, which is annoying at best, but we got a couple of things going for us:

1. Since n base k is straight up made of 1's, converting back to n from n base k yields a geometric sequence. Useful, as it simplifies calculation.

2. The biggest possible length n base k could have is n base 2, which is just $\log_2 n$

3. Every number has a small base, that being $n-1$, since $(n-1)^1+1=n$

3 Proof

This is a pure math problem, and we'll need to explain a few properties of the good base before we try to find it.

Firstly, we'll notice that any n can be represented from a geometric sequence. For instance, suppose we wanted to represent the decimal value of 11111 in base 2, which is:

$$1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 1 + 2 + 4 + 8 + 16 = 31$$

You might notice it looks exactly like a geometric sequence[1], which is just a set of numbers sharing a common ratio r which is used to get the next elements from a sequence. Fortunately for us, there is a formula to calculate the sum for the first m numbers in a geometric sequence:

$$S = a \cdot \frac{1 - r^m}{1 - r}$$

Therefore, for our good base numbers, we know the first element is *1* and the common ratio is the base k, so:

$$n = 1 \cdot \frac{1 - k^m}{1 - k} \tag{1}$$

We know n, we need to find out m and k.

Fortunately, m here is the number of digits the number made of solely 1's. This number has a maximum cap to its length, per note *2* above. Therefore, we can just go down from the maximum number of digits possible to 1 to deduce k. Since we now know both n and m, we can now deduce what we are looking for, which is k.

In the case that k and m are large numbers comparatively to 1, we can estimate the above equation to:

[1] For any element x_n, $x_n = a \cdot r^{n-1}$, where a is the first element, and r is the common ratio.

$$n \approx \frac{-k^m}{-k} \tag{2}$$
$$= \frac{k^m}{k} \tag{3}$$
$$= k^{m-1} \tag{4}$$

Pulling the root of $m-1$ on both sides, we get our estimation of k, which is:

$$k = \sqrt[m-1]{n} = n^{\frac{1}{m-1}}$$

Important to note the above is an estimation, which, in our case, will likely be accurate if rounded down to integers. Just to be safe, however, we'll check precisely using the geometric sequence formula to ensure we get n with all the variables we've figured out.

4 Solution

This was annoying, right? Well, thanks to the above, the code is short. We'll begin with the Leetcode boilerplate.

```
class Solution:
    def smallestGoodBase(self, n: str) -> str:
```

We'll convert the provided string to an integer and get the maximum value of m we needed earlier, which is just $\log_2 n$. We'll cast to an integer, since we need the number of digits that can represent n at maximum:

```
        n = int(n)
        m_max = int(math.log(n, 2))
```

Now we'll iterate through every single value of m possible. We'll exclude, since there are no numbers with 0 digits[2]:

```
        for m in range(m_max, 1, -1):
```

We'll get out approximation of k and then verify if it is the correct value. We'll be using the geometric sequence formula. Note we reversed the signs on the sides of the fraction for readability and moved the denominator to n, so this code will verify if:

[2] Also, there's the fact that we'll be dividing by m, which can't be zero.

$$k^(m+1)-1=n\cdot(k-1) \tag{5}$$

If that is the case, return the good base we have just found.

```
k = int(n**(1 / m))
if (k**(m + 1) - 1) == n * (k - 1):
    return str(k)
```

In the worst case, we haven't found any good bases with our approximation, we'll just output the biggest good base, which is just $n-1$, as per note 3.

```
return str(n - 1)
```

Time Complexity: $O(\log_2 n)$, as we're just running constant operations on every possible number of digits for n base k.