Project Euler Problem 686

 $2^7=128$ is the first power of two whose leading digits are "12". The next power of two whose leading digits are "12" is 2^{80} .

Define p(L, n) to be the nth-smallest value of j such that the base 10 representation of 2^j begins with the digits of L. So p(12, 1) = 7 and p(12, 2) = 80.

You are also given that p(123, 45) = 12710.

Find p(123, 678910).

Analysis

For given L, we iterate through k and check whether there is a power of 2 between $L \cdot 10^k$ and $(L+1) \cdot 10^k$, i.e., $\exists j \in \mathbb{Z}^{\geq 0} \ [L \cdot 10^k \leq 2^j < (L+1) \cdot 10^k]$. For each such k we find, we have a corresponding j such that the base 10 representation of 2^j begins with the digits of L. We thus need to find n such k's and return the value of j that corresponds with the last k.

If there is a power of 2 strictly between $L\cdot 10^k$ and $(L+1)\cdot 10^k$, we must have for $j\in\mathbb{Z}^{\geq 0}$

$$\begin{split} L \cdot 10^k < 2^j < (L+1) \cdot 10^k &\Longleftrightarrow \log_2(L \cdot 10^k) < j < \log_2((L+1) \cdot 10^k) \\ &\Longleftrightarrow \log_2 L + k \cdot \log_2 10 < j < \log_2(L+1) + k \cdot \log_2 10 \\ &\Longleftrightarrow \lfloor \log_2 L + k \cdot \log_2 10 \rfloor < \lfloor \log_2(L+1) + k \cdot \log_2 10 \rfloor \end{split}$$

The case where $L \cdot 10^k$ is itself a power of 2 is only possible when k=0, since powers of 2 always end in 2,4,6, or 8. We can therefore account for it simply by checking whether L is a power of 2.

Given k, the relationship $2^{j-1} < L \cdot 10^k \le 2^j < (L+1) \cdot 10^k$ implies $j = \lceil \log_2 L + k \cdot \log_2 10 \rceil$.

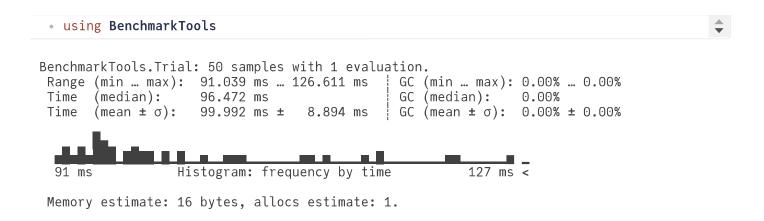
Solution

```
p (generic function with 1 method)
function p(L, n)
      # pre-computing logarithms
      log2_L, log2_Lp1, log2_10 = log2.((L, L+1, 10))
      i = isinteger(log2\_L) ? 1 : 0 # accounting for case L is a power of 2
      \mathbf{k} = 0
      while true
          # check for power of 2
          if floor(log2_L + k*log2_10) < floor(log2_Lp1 + k * log2_10)
              i += 1
          end
          i == n && break # break if sufficiently many powers found
          k += 1 # increment power of 10
      # compute corresponding exponent of 2
      return ceil(Int, log2_L + k*log2_10)
end
```

193060223

• p(L, n)

Benchmark



Validation

```
• @assert p(12, 1) == 7
```

```
• @assert p(12, 2) == 80
```

```
• Qassert p(123, 45) == 12710
```

```
• <code>@assert p(128, 1) == 7 # tests edge case where L is a power of 2</code>
```