# IBM Ponder - Dec, 2024

Mohit Garg

December 21, 2024

## 1 Problem Statement

We define $\omega_d(n)$ to be the number of integers $k$ in $\{1, 2, \ldots, n\}$ whose (usual decimal) representation contains the digit $d$. That is:

$$\omega_d(n) \;=\; \#\big\{\, k : 1 \le k \le n,\; k \text{ has digit } d \text{ somewhere}\big\}.$$

The ratio

$$\frac{\omega_d(n)}{n}$$

represents the probability (for an integer uniformly chosen from $\{1, \ldots, n\}$) of containing the digit $d$.

- **Main Goal:** For each $d \in \{1, 2, \ldots, 9\}$, find

$$\max\big\{\, n : \frac{\omega_d(n)}{n} = \tfrac{1}{2}\big\}.$$

- **Optional Goal:** Find as simple a formula as possible in $d$ for these maximum values.
- **Bonus Goal:** Find the unique maximal $n$ such that

$$\frac{\omega_1(n)}{n} \;=\; \frac{3}{4}.$$

## 2 Step 1: Brute Force Approach for the Main Goal

We start by noting that for very large $n$, the ratio $\omega_d(n)/n$ approaches 1. Hence if there is an $n$ such that $\omega_d(n)/n = 1/2$, it must be below a certain threshold (since eventually the ratio grows beyond $1/2$ and never returns).

### 2.1 Brute-Force Implementation for Smaller Ranges

A direct way to locate that threshold for each digit $d$ is to:

1. Iterate $n$ from 1 up to some high limit (e.g. $10^8$ or $10^9$).
2. For each $n$, compute $\omega_d(n)$ by checking every integer up to $n$ for digit $d$.
3. Identify the last $n$ for which $\omega_d(n)/n = 1/2$.

Although naive, this is feasible up to about $n \approx 10^8$ if implemented in a suitably optimized manner which we have done using cumulative counts (check Appendix).

### 2.2 Numerical Results Up to $10^8$

Using such a brute force count up to $10^8$, we found that for each $d \in \{1, 2, \ldots, 9\}$, the ratio $\omega_d(n)/n$ crosses $1/2$ at some point and then remains above $1/2$ once $n$ gets sufficiently large. Specifically, the *maximum* $n$ (below $10^8$) for which the ratio equals $1/2$ was determined as follows (listed as pairs $(d,\ n_{\max})$):

$$(1,\ 1062880),$$
$$(2,\ 2125762),$$
$$(3,\ 3188644),$$
$$(4,\ 4251526),$$
$$(5,\ 5314408),$$
$$(6,\ 6377290),$$
$$(7,\ 17006110),$$
$$(8,\ 18068992),$$
$$(9,\ 19131874).$$

Further checks (still by brute force) up to $n = 100$ million confirm that none of these values increases beyond the above points: past these $n_{\max}$, the ratio remains strictly above $1/2$.

**Bounding the Maximum $n$ Below $10^8$:** A simple bound for $\omega_1(n)$ near $10^k$ uses the approximation

$$1 - \left(\frac{9}{10}\right)^k,$$

so that, by $k = 7$, we already exceed $1/2$. Additionally, from Figure 1, we observe that even the troughs of the lines for the different digits do not cross the 0.5 barrier. This ensures that the maximum $n$ satisfying $\frac{\omega_d(n)}{n} = \frac{1}{2}$ does not exceed $10^8$.

### 2.3 Piecewise Linear Relationship in $d$

Plotting the maximum $n_{\max}(d)$ against $d$ (for $d = 1, 2, \ldots, 9$) reveals an unexpected *piecewise linear* pattern with a large jump between $d = 6$ and $d = 7$. A simple fitting to the points produced a function of the form:

$$f(d) \ = \ \begin{cases} 1062880 + 1062882\,(d-1), & \text{if } 1 \le d \le 6, \\ 1062880 + 1062882\,(d+8), & \text{if } 7 \le d \le 9. \end{cases}$$

Though numeric coincidence (like 1062882) appears, I was unable to uncover a deeper combinatorial explanation for it. Empirically, this function perfectly matches the brute-force findings. This can be seen in Figure 2.

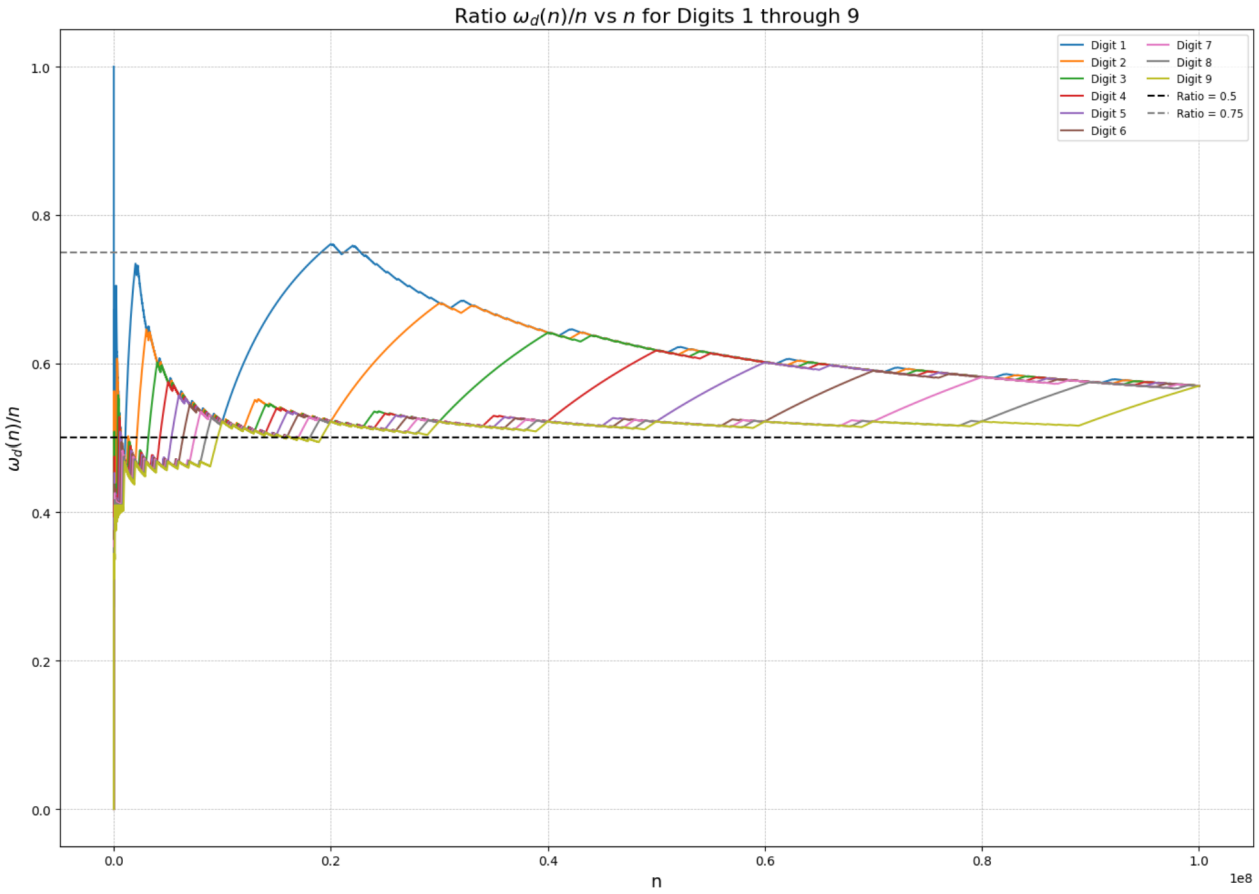


Figure 1: Ratio $\omega_d(n)/n$ vs. $n$ for $d = 1, \ldots, 9$ up to $n = 10^8$. The black dashed line is the ratio 0.5, and the upper gray dashed line is 0.75. We see each curve eventually stays above 0.5 after crossing it.

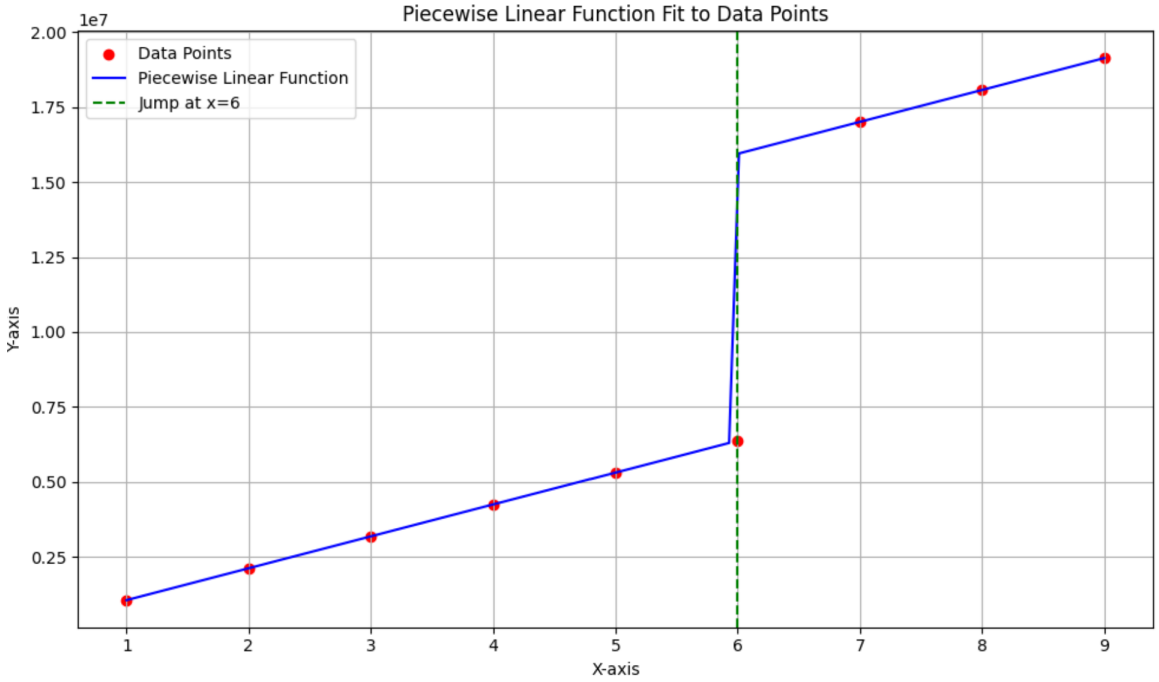


Figure 2: Piecewise linear fit to the maximum $n$ where $\omega_d(n)/n = 1/2$, showing a jump between $d = 6$ and $d = 7$. We keep $n$ in the range $[10^0, 10^8]$.

# 3 Step 2: Extending to the Bonus Goal $\omega_1(n)/n = 3/4$

The "bonus" ratio 3/4 is a more extreme threshold. We expect that it will be achieved at a much larger $n$ (around 13–14 digits), since asymptotically $\omega_1(n)/n \to 1$. Brute forcing up to $10^{14}$ or so is of course intractable on a normal machine.

## 3.1 Theoretical Bounds

We know that for $n = 10^k - 1$, the proportion of integers containing digit 1 is approximately

$$1 \; - \; \left(\tfrac{9}{10}\right)^k.$$

Hence once $k$ is large, this quantity approaches 1. Specifically, to exceed 3/4, we need

$$1 \; - \; \left(\tfrac{9}{10}\right)^k \; \geq \; \tfrac{3}{4} \quad \Longrightarrow \quad \left(\tfrac{9}{10}\right)^k \; \leq \; \tfrac{1}{4},$$

which gives $k \approx 13.4$. Thus $10^{14}$ is definitely *beyond* the point at which $\omega_1(n)/n$ will exceed 3/4 and never return. So we only need to search up to around $10^{14}$ or $10^{15}$.

## 3.2 Digit-DP + Binary Search

To *exactly* locate the unique integer $n_{\max}$ with $\omega_1(n_{\max})/n_{\max} = 3/4$, we adopt:

1. **Digit-DP:** A function $\rho_1(N)$ counts how many integers in $[0, N]$ do *not* contain digit 1. This can be done by a standard dynamic-programming over the decimal digits of $N$. The complexity is on the order of $O(\text{number of digits} \times 10)$, extremely fast for up to 14 or 15 digits.

2. **Binary Search:** We do a binary search in $[1, 10^{15}]$ (or $[1, 10^{14}]$, depending on how conservative we want to be) looking for the largest $n$ such that

$$\frac{\omega_1(n)}{n} \; = \; \frac{3}{4}.$$

   In integer form, $\omega_1(n) = \frac{3}{4}\, n$ translates to

$$\omega_1(n) \; = \; (n+1) - \rho_1(n),$$

   where $\rho_1(n)$ is the count of numbers in $[0, n]$ missing digit 1. We check if

$$4 \times \omega_1(n) \; = \; 3 \times n \quad \Longleftrightarrow \quad 4\left[(n+1) - \rho_1(n)\right] = 3\, n.$$

   Because $\rho_1(n)$ can be computed in $O(\log n)$ time, and the binary search does $O(\log n)$ iterations, the entire procedure is efficient.

## 3.3 Final Answer for 3/4

Using this algorithm yields

$$n_{\max} \; = \; 1{,}129{,}718{,}145{,}920 \quad \text{such that} \quad \frac{\omega_1(n_{\max})}{n_{\max}} \; = \; \frac{3}{4}.$$

Once $n$ exceeds this value, the ratio remains strictly above 3/4.

## 4 Why $n \leq 10^{15}$ Suffices

As reasoned, the ratio approaches 1 as $n \to \infty$. Once it climbs above 3/4, it never dips back down (at large scales). A simple bound for $\omega_1(n)$ near $10^k$ uses the approximation

$$1 \; - \; \left(\frac{9}{10}\right)^k$$

so that, by $k = 14$, we already exceed 3/4. Thus $n = 10^{14}$ is an ample upper bound for searching the exact crossing. We extended to $10^{15}$ just for completeness.

## 5 Conclusion

**Main Goal (ratio = 1/2):** We confirmed (via brute force up to $10^8$) that for each digit $d$, there is a unique maximal $n_{\max}(d)$ with

$$\frac{\omega_d(n_{\max}(d))}{n_{\max}(d)} = \frac{1}{2},$$

and no larger $n$ attains exactly 1/2. The values are:

$$\begin{aligned}
&(d = 1)\ 1062880, \quad (d = 2)\ 2125762, \quad (d = 3)\ 3188644, \quad (d = 4)\ 4251526,\\
&(d = 5)\ 5314408, \quad (d = 6)\ 6377290, \quad (d = 7)\ 17006110,\\
&(d = 8)\ 18068992, \quad (d = 9)\ 19131874.
\end{aligned}$$

They exhibit an intriguing piecewise linear relationship with a large jump between $d = 6$ and $d = 7$.

**Bonus Goal (ratio = 3/4):** We used a *digit-DP* approach to count $\rho_1(n)$, the number of integers in $[0, n]$ that omit the digit 1, in $O(\log_{10} n)$ time, and performed a binary search in the interval $[1, 10^{15}]$. The unique maximal $n$ for which

$$\frac{\omega_1(n)}{n} \; = \; \frac{3}{4}$$

is

$$1{,}129{,}718{,}145{,}920.$$

Beyond that $n$, the ratio remains above 3/4 indefinitely.

## A Appendix

Find the code used in this document at the following GitHub repository:
`https://github.com/Sp1cyJu1ce/IBM-Ponder-Solutions/tree/main/December2024`