

Analysis: Overrandomized

Test Set 1

In Test Set 1, the range for possible M values is so small compared to the number of records that each combination (M_i, N_i) has a somewhat large probability $1 / (99 \times M_i)$ of appearing as a particular record, and an even larger probability of being present as at least one record.

Suppose there exists at least one record with $M_i = N_i = x$ for each x in the range 1 through 9. From the record with $M_i = N_i = 1$ we know that the only letter in R_i represents 1. Then, from all the records with $M_i = 2$, we can discard the ones where R_i represents 1. The leftovers must be the ones with $M_i = N_i = 2$, and in those, the only letter in R_i represents 2. In general, after we have decoded the letters for 1 through x , we can take the records with $M_i = x + 1$ and discard the ones with letters in R_i that are already assigned, and the R_i values of the remaining records will contain the letter that should be assigned to $x + 1$. Finally, the only remaining unassigned letter should be assigned to 0.

This process works as long as records with $M_i = N_i = x$ exist for each x in the range 1 through 9. The probability of that happening is hard to calculate, but the least likely of those combinations is $M_i = N_i = 9$, with a probability of only $1 / (99 \times 9)$ per record. The probability of that appearing at least once in 10000 records is greater than 99.999%. The smaller values of x have even higher probability. Of course, the probability of all 9 coexisting is smaller than that, and the probability of that happening in all 10 cases is even smaller, but still decent enough. In addition, there is a really small probability that the letter representing 0 doesn't appear at all in the input, but if that were the case, no algorithm could find it. Given that this is a Visible Verdict test set, we can give the solution a try, and confirm that it passes.

There are additional heuristics we could add to the algorithm. For example, if we can't find the value for 9 in this way and we need to distinguish the two remaining letters to assign to 9 and 0, just having a record whose R_i starts with one of the letters is enough to know that that one should be a 9, since 0 cannot be a leading digit. This further increases the probability of the method working. We can add more and more heuristics to cover the remaining cases, but at some point it's easier to just try something more general.

Test Set 2

In Test Set 2, the probability of M_i being a single digit is small, and we cannot rely on that happening, let alone several times and with extra conditions. However, we can treat records that have M_i and R_i of the same length similarly, by simply using their first letters and then using those (digit, letter) pairs as we did in the solution for Test Set 1.

Test Set 3

At this point, it seems like we may have to throw all of the above insights away, because they are all predicated on knowing M_i . However, the "use the leading digit/letter" insight we used to solve Test Set 2 is actually the first step toward solving Test Set 3 as well.

In Test Set 3, each record's information comes from a single integer, not two, so we cannot use the association between the two parts as before. We can start by checking the distribution used

to generate the only piece of information we have. The probability of any particular N_i being equal to x is the sum of $10^{-16} / y$ for y in $[x, 10^{16} - 1]$, which can be approximated by $10^{-16} (\ln 10^{16} - \ln x)$. In other words, the probability is decreasing in x . Because of this, leading digits are more likely to be smaller than larger. Moreover, even though the decrease in probability between the actual result being x and $x + 1$ is small, the decrease in probability between the leading digit being d and $d + 1$ is large, because it aggregates the differences between the probability of dS being more than the probability of $(d+1)S$ for each possible suffix S . This is a version of [Benford's law](#).

Therefore, a possible solution is to calculate the frequency with which each letter appears as a leading digit, and assign the highest frequency to 1, the second highest to 2, etc. The only letter that never appears as a leading digit, and therefore has the minimum frequency, should be assigned to the digit 0.