Hidden Digits

Идея: Михаил Дворкин
Разработка: Николай Будин

Давайте реализуем рекурсивную функцию `solve(a, f)`, которая принимает массив двоичных масок $a_0, a_1, \ldots a_{k-1}$ и находит минимальное x такое, что в x встречаются все цифры из a_0, в $(x+1)$ встречаются все цифры из a_1, и т.д. Дополнительно эта функция будет принимать флаг f, обозначающий, может ли x равняться 0, или оно должно быть строго положительным.

Теперь рассмотрим три случая для функции `solve`:

- Если $|a| = 1$. Тогда нам нужно просто собрать минимальное число, которое содержит все цифры из a_0 и удовлетворяет флагу f. В большинстве случаев достаточно взять цифры из a_0 в возрастающем порядке, и если на первом месте оказался 0, то поменять местами первую и вторую цифры. Однако, есть крайние случаи, которые нужно аккуратно разобрать: если a_0 пусто; если x не может быть 0; если a_0 содержит только 0.
- Если $|a| = 2$. Тогда есть два интересных варианта, как могут выглядеть x и $x+1$:
  1. $x = \overline{\ldots d}$, $x + 1 = \overline{\ldots (d+1)}$, $d < 9$. То есть, x и $x+1$ имеют одинаковый префикс, и разные цифры на конце. Переберем d, вычеркнем d из a_0, вычеркнем $d+1$ из a_1, сделаем $b = a_0 | a_1$, и найдем минимальное число, содержащее все цифры из b.
  2. $x = \overline{\ldots d9}$, $x + 1 = \overline{\ldots (d+1)0}$, $d < 9$. Снова переберем d.
- Если $|a| > 2$. Тогда переберем $x \bmod 10$, то есть последнюю цифру первого числа. В результате, мы узнаем последние цифры всех чисел. Вычеркнем их из масок. Теперь отбросим у каждого числа последнюю цифру, то есть нацело поделим все числа на 10. В результате у нас будут отрезки подряд идущих одинаковых чисел. Объединим числа внутри такого отрезка в одно, при этом множества необходимых цифр объединятся. Рекурсивно вызовем `solve` от нового множества масок, чтобы найти минимальное подходящее значение $\lfloor \frac{x}{10} \rfloor$.

Пусть функция `solve` при $|a| = n$ работает за $T(n)$. Тогда $T(n) = 10 \cdot T(n/10) + O(n)$. Поэтому, $T(n) = O(n \log n)$.

Чтобы решить задачу, нужно запустить `solve` с $a_i = \{d_i\}$.