АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
МОНО

„Лакомият” алгоритъм, който със сигурност ще даде едно сравнително добро (не много различно от оптималното) решение е ясен: ако n е моно, то е и решението на задачата; иначе намираме най-голямото моно m, по-малко от n, запомняме го и решаваме задачата за n-m. Този алгоритъм може да дава и оптимално решение за много числа n, но такива тестове специално са избегнати.

Стандартното динамично решение се основава на най-естествените разсъждения:

- ако n е моно, то е и решение;
- иначе решението за n е минималното от обединението на решенията за i и n-i, където i обхожда интервала от 1 до [n/2].

Сложността на такова решение е квадратична. Нуждата от памет може да се намали много, ако се съобрази, че едно моно с до 15 цифри може да се опише, всъщност, в един байт: четири бита за цифрата и четири за броя цифри (всъщност, в един байт могат да се кодират и повече монота, чрез цялата част и остатъка по модул 9, но достъпът до свойствата „брой цифри” и „цифра” се забавя от делението, а и няма особен смисъл от така кодирани по-големи монота: дори линеен алгоритъм няма да мине в „нормално” време за такива големи числа). За подобно решение са предвидени 20% от точките.

За пълно решение на задачата правим следните разсъждения:

Можем да запишем всяко c-цифрено моно като $m_c = d\dfrac{10^c - 1}{9}$, където d е цифра, по-голяма от нула, а c е естествено число. Тогава търсеното представяне на n има вида:

$n = \sum_{i=1}^{p} m_{c_i} = \sum_{i=1}^{p} d_i \dfrac{10^{c_i} - 1}{9} = \dfrac{1}{9}\left(\sum_{i=1}^{p} (d_i \times 10^{c_i}) - \sum_{i=1}^{p} d_i \right)$ и искаме да минимизираме p.

Горното равенство да запишем като $9n + \sum_{i=1}^{p} d_i = \sum_{i=1}^{p} (d_i \times 10^{c_i})$ ..... (1)

Тъй като c_i са естествени, дясната страна се дели на 10. Така че търсим най-малкото p, за което:

- $a = 9n + s$, където s е естествено число, е кратно на 10;
- s се представя като сума от p на брой ненулеви цифри d_i (за i=1, 2, …, p);
- съществуват p степени на десетката c_i (за i=1, 2, …, p), за които $9n + s = \sum_{i=1}^{p} (d_i \times 10^{c_i})$.

В разглежданото представяне (1) се открояват две много различни ситуации:

1. Ако всички c_i са различни. Тогава дясната страна в (1) представлява точно единственото представяне на числото a в десетична бройна система. Значи, ако a е (p+1)-цифрено и сумата от цифрите на a е точно s, лесно можем да намерим d_i – това са просто ненулевите цифри на a, в същия ред.

2. Има равни c_i. Идеята е този случай да приведем до по-простия случай 1, като отстраним всевъзможните комбинации от равни степени.

Ясно е, че ако в (1) има t на брой равни степени, без ограничение на общността можем да считаме, че поне t-1 от тях имат съответна цифра 9. Наистина, ако допуснем, че пред две от тях цифрите d_1 и d_2 са различни от 9, то, ако $d_1 + d_2 < 10$, можем да намалим броя на събираемите, като заместим двете с едно, равно на тяхната сума. При $d_1 + d_2 \geq 10$, можем да заменим, например, d_1 с 9, а d_2 с $d_1 + d_2$ - 9, с което сумата не се променя, но едното от събираемите вече има съответна цифра 9.

Ще забележим, че няма смисъл да търсим представяне с повече от едно моно 9, защото вече 9+9=18=11+7 има минимално представяне от първия вид (с различни степени).

Оказва се, че това може да се обобщи: няма смисъл да търсим повече от две монота 99, защото 99+99+99=3×99=297=9+66+222.

Съвсем аналогични представяния съществуват за 4×999, 5×9999, … , 8×9999999. Представянето на 9×99999999=9+99+999+9999+99999+999999+9999999+888888888 всъщност е последното от тази редица. За 10×999999999, въпреки очакванията, няма такова представяне „от простия вид", то съществува чак за 11×999999999. Нататък, обаче, правилото продължава.

В (1) сумата от цифрите на a не може да надвишава s. Нека съберем цифрите пред някоя от равните степени отдясно (ако изобщо има такива) и получим w. Както казахме, има смисъл да разглеждаме равни степени само ако $w \geq 10$, иначе тези събираеми се редуцират в едно. В съответния разред ще остане цифрата $w \bmod 10$, а частното $[w / 10]$ ще се добави към по-старшия разред. Така сумата на цифрите вдясно ще нарасне най-много с $(w \bmod 10)+[w / 10]$ (и то, ако не се налага пренос към още по-старши разред), а тази сума е по-малка от w за всяко $w \geq 10$. Можем да обобщим: ако сумата на цифрите на a е по-малка от s, непременно има повтарящи се степени в (1).

От горните разсъждения можем да изведем следния алгоритъм:

1. Намираме едно представяне на n като сума от монота, с което получаваме една горна граница за търсения минимум p. За целта можем да ползваме първото (и единствено) p, отговарящо на (1), ако сумата от цифрите на a е точно s, или, в противен случай, „лакомия" алгоритъм, описан най-горе.
2. Намираме (ако не сме още) първото s, за което $a = 9n + s$ е кратно на 10 и сумата от цифрите на a не надхвърля s.
3. Ако сумата от цифрите на a е точно s, към 4, иначе към 5.
4. Определяме q като броя на ненулевите цифри в a; Ако $q < p$, полагаме $p=q$ и намираме монотата в представянето на n чрез цифрите на a. Преминаваме към 5.
5. Решаваме задачата за числата n-(99), n-(999), n-(99+999), … В скобите са всевъзможните комбинации от монота с цифра 9, като горните граници за всяка позиция се определят по съображенията, описани по-горе. Ако резултатът q за някое от тях е по-малък от p-(броя на девет-монотата), полагаме $p = q$+(броя на девет-монотата) и към неговия набор от събираеми монота добавяме съответните монота от деветки.

   В тази комбинаторна част от алгоритъма, ако генерираме наборите в (обратен) азбучен ред, имаме две ползи:

- сумите от девет-монота растат монотонно, което позволява спиране на процеса в първия момент, когато се надхвърли n;
- можем да „прескочим" много „невъзможни" (поради броя събираеми) комбинации, като генерираме следващата в (обратен) азбучен ред, която е възможна, без опасност да пропуснем някой случай.

6. Извеждаме p и получения набор от монота.

Автор: Павлин Пеев