Informatyczny kącik olimpijski (77): Wybrakowany ciąg

W tym miesiącu omówimy zadanie *Sequence*, które pojawiło się na tegorocznej edycji Bałtyckiej Olimpiady Informatycznej. Na tablicy napisano n kolejnych liczb całkowitych $p, p+1, \ldots, p+n-1$. Następnie z każdej liczby usunięto wszystkie cyfry oprócz jednej. Znając ciąg n cyfr $d_0, d_1, \ldots, d_{n-1}$, które pozostały na tablicy, należy wyznaczyć minimalną wartość p, od której mógł zaczynać się pierwotny ciąg.

Przykładowo, ciąg cyfr $2, 8, 4, 3, 1$ mógł powstać z pięciu kolejnych liczb naturalnych w następujący sposób: $4\underline{2}7, 42\underline{8}, \underline{4}29, 4\underline{3}0, 43\underline{1}$.

Zacznijmy od prostej obserwacji, że dla każdego ciągu cyfr szukana liczba p istnieje. Wystarczy zauważyć, że dla $p = 1234567890 \cdot 10^{\lceil \log_{10} n \rceil}$ wszystkie liczby z ciągu $p, p+1, \ldots, p+n-1$ zaczynają się od prefiksu zawierającego wszystkie możliwe cyfry.

Zadanie jest niełatwe i tylko jeden z uczestników olimpiady zdobył za nie maksymalną liczbę punktów. Kluczowe dla rozwiązania jest przyjrzenie się cyfrze jedności liczby p; oznaczmy ją przez x. Wyznacza ona jednoznacznie cyfry jedności pozostałych liczb pierwotnego ciągu – będą to kolejno

$$x_0 = x, \quad x_1 = (x+1) \bmod 10, \quad \ldots, \quad x_{n-1} = (x+n-1) \bmod 10.$$

Jeśli $x_i = d_i$, to pozostałe cyfry i-tej liczby mogą być dowolne. W przeciwnym przypadku cyfra d_i musi się pojawić na dalszej pozycji w i-tej liczbie. Podzielmy teraz poszukiwany ciąg na bloki zaczynające się na takich pozycjach, że $i = 0$ lub $x_i = 0$. Każdy blok (oprócz pierwszego i ostatniego, które mogą być krótsze) zawiera 10 liczb, które w pierwotnym ciągu będą się różnić jedynie na cyfrze jedności. Zatem ich wspólny prefiks będzie musiał zawierać wszystkie cyfry ze zbioru $\{d_j \mid i \leqslant j < i+10,\ x_j \neq d_j\}$, gdzie i jest początkową pozycją bloku. Możemy więc rozważyć 10 razy krótszy ciąg – jego elementy będą zbiorami cyfr, które muszą wystąpić w kolejnych blokach, i powtórzyć powyższe rozumowanie.

Zatem zadanie rozwiązujemy rekurencyjnie: dla danego n-elementowego ciągu $D_0, \ldots, D_{n-1}$, zawierającego zbiory cyfr, które mają wystąpić w kolejnych liczbach, należy wyznaczyć minimalne p. Jeśli $n = 1$, to odpowiedzią jest liczba zawierająca wszystkie cyfry ze zbioru D_0 (w kolejności rosnącej). Dla $n = 2$ mamy dwa przypadki: albo $x \neq 9$ i wtedy liczby mają ten sam prefiks, więc rekurencyjnie znajdujemy rozwiązanie dla zbioru cyfr $(D_0 - \{x\}) \cup (D_1 - \{x+1\})$, albo $x = 9$ i rekurencyjnie rozpatrujemy ciąg $D_0 - \{9\}, D_1 - \{0\}$, wiedząc, że nie opłaca nam się więcej rozpatrywać drugiego przypadku. Zatem rozwiązanie dla ciągów długości $n \leqslant 2$ znajdujemy w czasie stałym.

W końcu dla $n \geqslant 3$ przeglądamy wszystkich 10 kandydatów x na cyfrę jedności p. Dla każdego z nich budujemy ciąg o długości $n' \leqslant \lfloor \frac{n}{10} \rfloor + 2 < n$, zawierający sumy zbiorów dla kolejnych bloków, i rekurencyjnie wyznaczamy dla niego wartość $p'(x)$. Szukane p to minimum ze zbioru $\{10 \cdot p'(x) + x \mid 0 \leqslant x \leqslant 9\}$.

Złożoność czasowa algorytmu wyraża się wzorem rekurencyjnym

$$T(1) = T(2) = O(1), \quad T(n) = 10 \cdot \left(T\left(\lfloor \tfrac{n}{10} \rfloor + 2\right) + O(n)\right),$$

którego rozwiązaniem jest $T(n) = O(n \log n)$.

Niestety, powyższy algorytm ma dość istotny feler – działa przy założeniu, że liczby w pierwotnym ciągu mogą mieć zera wiodące (np. dla $n = 1$ i zbioru $D_0 = \{0, 1, 3, 7\}$ konstruujemy $p = 0137$ zamiast poprawnego $p = 1037$). I, jak to zwykle bywa w zadaniach tego typu, naprawienie tej usterki wymaga dość żmudnego przyjrzenia się poszczególnym przypadkom, w którym takie zero może się pojawić. W szczególności, gdy cyfra jedności jednej z liczb należącej do bloku jest „istotnym” zerem (czyli $x_i = 0$ oraz $0 \in D_0$), wygodnie jest dorzucić do zbioru D odpowiadającego temu blokowi dodatkowy element sygnalizujący, że w p musi wystąpić jakaś niezerowa cyfra. Szczegóły techniczne pozostawimy do dopracowania Czytelnikom.

Tomasz IDZIASZEK

Rozwiązanie zadania F 867.
W warunkach zadania masa wody dm parująca w bardzo krótkim czasie dt jest proporcjonalna do pola powierzchni wody S, czyli $\mathrm{d}m = \alpha \cdot S\mathrm{d}t$, gdzie α jest współczynnikiem proporcjonalności. Zmiana poziomu wody w akwarium dh wiąże się ze zmianą jej masy zależnością $\mathrm{d}m = \rho \cdot S\mathrm{d}h$, gdzie ρ – gęstość wody. Stąd $\mathrm{d}h = (\alpha/\rho) \cdot \mathrm{d}t$. Tak więc w stałych warunkach parowania zmiana poziomu wody jest proporcjonalna do czasu. Jeżeli więc przez dwie doby poziom wody obniżył się o 1 centymetr, to cała woda, przy głębokości akwarium równej 15 cm, wyparuje po 30 dobach.