Informatyczny kącik olimpijski (132): *XORanges*

W tym odcinku omówimy rozwiązanie zadania XORanges, które pojawiło się na trzecich zawodach European Junior Olympiad in Informatics (Maribor, Słowenia).

XORanges: *Dany jest ciąg n liczb naturalnych $a = (a_1, a_2, \ldots, a_n)$ oraz q poleceń dwóch typów. 1) Zmień wartość i-tego elementu. 2) Podaj współczynnik przedziału $a_x, a_{x+1}, \ldots, a_y$. Współczynnik przedziału $a_x, a_{x+1}, \ldots, a_y$ obliczamy w następujący sposób: dla każdego podsłowa (spójnego fragmentu) tego przedziału liczymy xor jego wartości, następnie xorujemy te wyniki, otrzymując współczynnik przedziału. Przykładowo współczynnik przedziału (a_3, a_4, a_5) wynosi $a_3 \oplus a_4 \oplus a_5 \oplus (a_3 \oplus a_4) \oplus (a_4 \oplus a_5) \oplus (a_3 \oplus a_4 \oplus a_5)$. Zaproponuj algorytm, który jest w stanie możliwie szybko wykonać wszystkie polecenia podane na wejściu.*

Na początku przyjmijmy, że polecenia typu 1) będziemy wykonywali w czasie $O(1)$ poprzez zmianę wartości a_i. Zatem zajmijmy się odpowiedziami na polecenia typu 2). Załóżmy, że chcemy obliczyć współczynnik $a_x, a_{x+1}, \ldots, a_y$. Na potrzeby artykułu wprowadźmy kilka oznaczeń. Niech:

- $[a_i; a_j]$ oznacza ciąg $(a_i, a_{i+1}, \ldots, a_{j-1}, a_j)$,
- „*xor* podsłowa $[a_i; a_j]$" oznacza wartość $a_i \oplus a_{i+1} \oplus \ldots \oplus a_{j-1} \oplus a_j$.

Rozwiązanie $O(q \cdot n^3)$
Pierwsze rozwiązanie polega na obliczeniu *xora* każdego z $\frac{d(d-1)}{2}$ podsłów, gdzie $d = y - x + 1$ i oznacza długość przedziału $[x; y]$. Obliczenie *xora* podsłowa odbywa się za pomocą metody naiwnej, polegającej na przejściu po wszystkich elementach. Na koniec *xorujemy* powyższe wyniki. Rozwiązanie dla jednego polecenia działa w czasie $O(n^3)$, zatem całkowita złożoność to $O(q \cdot n^3)$.

Rozwiązanie $O(q \cdot n^2)$
W tym rozwiązaniu, podobnie jak wyżej, obliczymy *xor* podsłów $[a_{x'}; a_{y'}]$ (dla wszystkich $x \leqslant x' \leqslant y' \leqslant y$) niezależnie. Jednak tym razem, zamiast naiwnej metody, skorzystamy z obserwacji, że:

$$a_{x'} \oplus a_{x'+1} \oplus \ldots \oplus a_{y'} = s_{y'} \oplus s_{x'-1},$$

gdzie $s_i = a_1 \oplus a_2 \oplus \ldots \oplus a_i$. Ciąg $s = (s_0, s_1, s_2, \ldots, s_n)$ ma analogiczną konstrukcję do ciągu sum prefiksowych i możemy go wygenerować w czasie $O(n)$. Otóż $s_0 = 0$, zaś $s_i = s_{i-1} \oplus a_i$ (dla $i > 0$). W ten sposób otrzymaliśmy rozwiązanie działające w czasie $O(q \cdot n^2)$.

Rozwiązanie $O(q \cdot n)$
Przypomnijmy proste fakty:

- $g \oplus g = 0$ (dla g naturalnego),
- $g \oplus 0 = g$ (dla g naturalnego),
- $\oplus$ jest operacją łączną i przemienną.

Na podstawie powyższych faktów możemy zauważyć, że do obliczenia współczynnika przedziału wystarczy znać parzystość liczby wystąpień każdego elementu w wyrażeniu je opisującym. Przykładowo:

$$a_3 \oplus a_4 \oplus a_5 \oplus (a_3 \oplus a_4) \oplus (a_4 \oplus a_5) \oplus (a_3 \oplus a_4 \oplus a_5) = \\ = a_3 \oplus a_5,$$

gdyż a_3 i a_5 występują nieparzyście wiele razy w całym wyrażeniu, zaś a_4 występuje parzyście wiele razy.

Zastanówmy się, ile razy a_i (dla $x \leqslant i \leqslant y$) występuje we wzorze opisującym współczynnik przedziału $[a_x; a_y]$. Innymi słowy chcemy obliczyć, w ilu podsłowach występuje a_i. Niech f_i oznacza tę wartość. Otóż początkiem takiego podsłowa może być element o numerze z przedziału $[x; i]$, końcem zaś element o numerze z przedziału $[i; y]$.

$$\underbrace{a_x, \ldots, a_{i-1}, a_i}_{i-x+1}\!\!\!\!\!\overbrace{, a_{i+1}, \ldots, a_y}^{y-i+1}$$

Zatem $f_i = (i - x + 1)(y - i + 1)$. Współczynnik przedziału to *xor* zbioru $\{a_i \mid x \leqslant i \leqslant y \text{ i } f_i \text{ jest nieparzyste}\}$. Wyznaczenie tego zbioru realizujemy w czasie $O(n)$. Zatem całe rozwiązanie działa w czasie $O(q \cdot n)$.

Rozwiązanie $O(n + q \cdot \log(n))$
Przed nami rozwiązanie wzorcowe. Polecenie 1) będziemy wykonywali nieco inaczej niż poprzednio, ale najpierw opiszemy operację 2).

Przeanalizujmy parzystości elementów ciągu f. Otóż f_i jest nieparzyste, jeśli $(i - x + 1)$ i $(y - i + 1)$ są nieparzyste. Stąd f_i jest nieparzyste wtedy i tylko wtedy, gdy x, y, i są tej samej parzystości. Zatem jeśli x i y są różnej parzystości, to odpowiedzią jest 0. W przeciwnym przypadku odpowiedzią jest:

$$a_x \oplus a_{x+2} \oplus a_{x+4} \oplus \ldots \oplus a_{y-2} \oplus a_y.$$

Chcemy szybko znajdować wartość powyższego wyrażenia. Zauważmy, że jest to *xor* elementów na pozycjach parzystych lub nieparzystych ciągu $[a_x; a_y]$. Niech zatem:

- $a^N = (a_1, 0, a_3, 0, a_5, 0, a_7, 0, \ldots)$,
- $a^P = (0, a_2, 0, a_4, 0, a_6, 0, a_8, \ldots)$.

Wówczas dla x parzystego odpowiedzią jest $a^P_x \oplus a^P_{x+1} \oplus \ldots \oplus a^P_y$, zaś dla x nieparzystego odpowiedzią jest $a^N_x \oplus a^N_{x+1} \oplus \ldots \oplus a^N_y$.

Zbudujmy dwa drzewa przedziałowe. Liśćmi pierwszego z nich niech będą wartości ciągu a^N, zaś liśćmi drugiego niech będą wartości ciągu a^P. Pozostałe węzły będą przechowywały *xor* wartości dzieci. Taka struktura pozwala w czasie $O(\log(n))$ znajdować *xor* przedziału oraz w tym samym czasie obsługiwać operację zmiany wartości elementu (polecenie typu pierwszego). Zatem całe rozwiązanie działa w czasie $O(n + q \cdot \log(n))$.

Bartosz ŁUKASIEWICZ