Zadanie: NAW
Gra w życie

Warsztaty ILO 2017-2018, grupa olimpijska, dzień 1. Dostępna pamięć: 128 MB.

Rozwiązanie wzorcowe $O(n\ log\ n)$

Jeżeli rozważamy ciąg składający się z samych zer, to jesteśmy już w stanie końcowym i możemy zwrócić `NIE`. W przeciwnym wypadku zauważmy, że ciąg musi być nieparzystej długości, jeśli symulacja ma się zakończyć. Jest tak dlatego, że gdybyśmy próbowali wykonać symulację w drugą stronę, (tj. zaczynamy z samymi zerami i zastanawiamy się, jak ciąg wyglądał wcześniej), to już w pierwszym kroku dojdziemy do sprzeczności, jeśli przyjmiemy, że ciąg ma parzystą długość.

W związku z tym został nam już tylko przypadek, kiedy ciąg ma długość nieparzystą i nie składa się z samych zer.

Obserwacja.1. *Wartość i-tego pola w pewnym kroku symulacji jest wynikiem operacji $\oplus$ (xor) pól na pozycjach $i-1$ oraz $i+1$ ciągu z poprzedniego kroku symulacji.*

Obserwacja.2. *Pozycje parzyste i nieparzyste są w pewnym sensie niezależne. Pozycje nieparzyste generują nam wartości pozycji parzystych w kolejnym kroku, a pozycje parzyste generują wartości pozycji nieparzystych w kolejnym kroku.*

Powyższa obserwacja oznacza, że możemy niezależnie sprawdzić, czy podciąg składający się z pozycji parzystych zbiegnie kiedykolwiek do ciągu zerowego oraz analogicznie podciąg składający się z pozycji nieparzystych.

Rozważmy ciąg $x_1, x_2, \cdots, x_k$, będący podciągiem na pozycjach nieparzystych. Wówczas w kolejnym kroku otrzymamy ciąg: $y_1, y_2, \cdots, y_{k-1}$, który jest podciągiem na pozycjach parzystych ciągu otrzymanego w kolejnym kroku symulacji. Ponadto zauważmy, że $y_i = x_i \oplus x_{i+1}$. Zastanówmy się natomiast, jak będzie wyglądał ciąg $z_1, z_2, \cdots, z_k$, będący z kolei podciągiem pozycji nieparzystych po dwóch krokach symulacji. Otrzymujemy $z_i = y_{i-1} \oplus y_i = (x_{i-1} \oplus x_i) \oplus (x_i \oplus x_{i+1}) = x_{i-1} \oplus x_{i+1}$. A zatem okazuje się, że po skompresowaniu jednego kroku symulacji do dwóch dla pozycji nieparzystych, otrzymujemy dokładnie taki sam problem jak w oryginalnym zadaniu, co oznacza, że możemy rekurencyjnie sprawdzić, czy podciąg złożony z pozycji nieparzystych jest zbieżny do ciągu zerowego.

Co w przypadku podciągu składającego się z pozycji parzystych? Sytuacja niewiele się komplikuje, ponieważ możemy wykonać jeden krok symulacji, otrzymując pozycje nieparzyste, a stąd możemy już kontynuować powyższe rozumowanie, czyli uruchomić nasze rozwiązanie rekurencyjnie dla otrzymanego ciągu.

A zatem rozwiązanie możemy przedstawić następująco:

```
wczytaj_an

def zbiega(an)
    if ciag_zerowy(an)
        return True
    if n mod 2 == 0
        return False

    a_parz = pozycje_parzyste(an)
    a_nparz = pozycje_nieparzyste(an)

    return zbiega(a_nparz) && zbiega(krok(a_parz))

wypisz(zbiega(an))
```

Złożoność rozwiązanie jest $O(n\ log\ n)$, ponieważ funkcja `zbiega` wywołuje się rekurencyjnie na dwóch ciągach o połowie krótszych od wejściowego. Oznacza to, że każdy element ciągu zostanie przetworzony $O(logn)$ razy. Ponadto sprawdzenie, czy ciąg jest ciągiem samych zer oraz wykonanie jednego kroku symulacji jest bardzo proste i wymaga $O(n)$ operacji.