

Zadanie: DYN

Dynamiczny ciąg

Laboratorium z ASD, lab 7. Dostępna pamięć: 128 MB.

15.12.2024, 23:59:59

Zaimplementuj strukturę danych, która wykonuje podstawowe operacje na ciągu liczb, w którym dopuszczamy wstawianie elementów. Elementy ciągu numerujemy od 0. Początkowo ciąg jest pusty. Dozwolone są następujące operacje:

- $insert(j, x, k)$ – wstawia k egzemplarzy elementu x tuż przed j -tą pozycją ciągu, przesuwając elementy znajdujące się na pozycjach j oraz dalszych o k pozycji w prawo;
- $get(j)$ – daje w wyniku wartość j -tego elementu ciągu.

Możesz założyć, że w przypadku operacji $insert$ zachodzi $0 \leq j \leq n$, a w przypadku operacji get $0 \leq j < n$, przy czym n oznacza długość ciągu w momencie wykonywania operacji.

W tym zadaniu format wejścia jest zakamuflowany tak, by wymuszać rozwiązania działające *on-line*. W swoim rozwiązaniu możesz posłużyć się gotową implementacją zrównoważonych drzew binarnych z Internetu.

Wejście

W pierwszym wierszu znajduje się liczba całkowita m ($1 \leq m \leq 500\,000$), oznaczająca liczbę operacji. Każdy z kolejnych m wierszy zawiera jedną małą literę oznaczającą typ operacji – i dla $insert$, g dla get – po której następują liczby oddzielone odstępami: j' , x oraz k dla $insert$, j' dla get . Parametr j w danym zapytaniu jest wyznaczany na podstawie wczytanej liczby j' według wzoru $j = (j' + w) \bmod (n + 1)$ w przypadku operacji $insert$ oraz $j = (j' + w) \bmod n$ w przypadku operacji get , przy czym w to wynik ostatniej operacji get lub 0, jeśli nie było wcześniej żadnej operacji get , a n to długość ciągu w momencie wykonywania operacji. Liczby j' oraz x będą nieujemne i nie większe niż 10^9 , a liczby k będą z zakresu od 1 do 1000.

Wyjście

Dla każdej operacji get Twój program powinien wypisać jeden wiersz zawierający wynik tej operacji. Możesz założyć, że zostanie zawsze wykonana co najmniej jedna operacja tego typu.

Przykład

Dla danych wejściowych:

```
7
i 0 2 3
i 1 1 2
g 2
i 4 1 1
g 2
i 1 3 2
g 2
```

poprawnym wynikiem jest:

```
1
2
3
```

Dane wejściowe tłumaczą się na następujący ciąg zapytań:

wczytana operacja	przetłumaczona operacja	wynikowy ciąg	wynik operacji
i 0 2 3	$insert(0, 2, 3)$	2, 2, 2	–
i 1 1 2	$insert(1, 1, 2)$	2, 1, 1, 2, 2	–
g 2	$get(2)$	bez zmian	$w = 1$
i 4 1 1	$insert(5, 1, 1)$	2, 1, 1, 2, 2, 1	–
g 2	$get(3)$	bez zmian	$w = 2$
i 1 3 2	$insert(3, 3, 2)$	2, 1, 1, 3, 3, 2, 2, 1	–
g 2	$get(4)$	bez zmian	$w = 3$