

Aufgabe Peru

C header: peru.h
C++ header: peru.h

Diesen Morgen hat Roxy N Käfer auf ihrem Schreibtisch gefunden. Diese Käfer sind von 0 bis $N - 1$ durchnummeriert und jeder Käfer i hat Stärke S_i . Roxy will die Käfer zerquetschen, so dass sie ihre Mathematik-Hausaufgaben erledigen kann. Um dies zu erreichen hat sie einen speziellen Handschuh gekauft mit welchem sie eine zusammenhängende Teilsequenz von K Käfern treffen kann. Falls Roxy sich Mühe E gibt dann werden die Käfer deren Stärke S_i kleiner gleich E ist zerquetscht, wobei alle anderen Käfer keinen Schaden davontragen. Die zerquetschten Käfer bleiben bei ihrer ursprünglichen Position auf dem Tisch. Roxy kann den Handschuh so oft benutzen wie sie will.

Roxy will wissen, ob du die kleinste gesamte Mühe berechnen kannst, die sie sich jeweils geben müsste, um die ersten i Käfer zu zerquetschen, für jedes $1 \leq i \leq N$. Da das zu viele Zahlen gibt hätte Roxy stattdessen gerne das Resultat des folgenden Ausdrucks: $X_0 \cdot 23^{N-1} + X_1 \cdot 23^{N-2} + \dots + X_{N-1}$ modulo $10^9 + 7$, wobei X_i die minimale gesamte Mühe ist, um die ersten $i + 1$ Käfer zu zerquetschen.

Implementierungsdetails

Die folgende Funktion ist zu implementieren:

```
int solve(int N, int K, int* S);
```

Diese Funktion wird **genau einmal** aufgerufen und muss das Resultat des obenstehenden Ausdrucks modulo $10^9 + 7$ zurückgeben. Die Parameter dieser Funktion sind N (die Anzahl Käfer), K (die Länge der zusammenhängenden Teilsequenzen die sie mit dem Handschuh treffen kann) und S (ein Array der Länge N , wobei S_i die Stärke von Käfer i ist).

Limits

- $1 \leq N \leq 2\,500\,000$
- $1 \leq K \leq N$
- $1 \leq S_i \leq 2\,000\,000\,000$

Teilaufgabe 1 (18 Punkte)

- $1 \leq N \leq 2\,000$

Teilaufgabe 2 (31 Punkte)

- $1 \leq N \leq 400\,000$

Teilaufgabe 3 (51 Punkte)

- Keine weiteren Einschränkungen.

Beispiel

| input | output |
|-------------------------|-----------|
| 8 3 3 2 9 8 7 11 3 4 | 720026253 |

Das Array X ist $\{3, 3, 9, 12, 12, 20, 23, 23\}$.