

Xp – descrierea soluției

Autor: Asist. univ. dr. ing. Mugurel Ionuț Andreica – Universitatea Politehnica din București

Soluția 1

Complexitate de timp: $O(N^2)$

Memorie: $O(1)$ sau $O(N)$

Punctaj: 30 puncte

Fiecare valoare $Prod[i]$ este calculată independent (în timp $O(N)$).

Soluția 2

Complexitate de timp: $O(N)$

Memorie: $O(N)$

Punctaj: 50 puncte

În cadrul acestei soluții se vor genera toate valorile $val[i]$ și se vor memora într-un vector. Apoi vom calcula valorile $PR[i] = (val[i] \cdot \dots \cdot val[N]) \bmod Q$, în timp $O(N)$ per total. Avem $PR[N] = val[N]$ și $PR[1 \leq i \leq N-1] = (val[i] \cdot PR[i+1]) \bmod Q$.

Apoi vom parcurge valorile $val[i]$ de la 1 la N, menținând pe parcurs produsul primelor $i-1$ elemente. Când ajungem la poziția i , fie PL acest produs. Atunci vom avea $Prod[i] = (PL \cdot PR[i+1]) \bmod Q$ (unde considerăm $PR[N+1] = 1$).

Astfel, putem calcula toate valorile $Prod[i]$ în timp $O(N)$ per total.

Soluția 3

Complexitate de timp: $O(N \cdot \log(N))$

Memorie: $O(\log(N))$

Punctaj: 100 puncte

Această soluție folosește tehnica de programare denumită *Divide et Impera*. Se va scrie o funcție recursivă `divide_et_impera(i, j, A[i], B[i], outProd)`, în care `outProd` va reprezenta produsul elementelor din afara intervalului de poziții $[i, j]$ (bineînțeles, modulo Q). Vom efectua primul apel `divide_et_impera(1, N, A[1], B[1], 1)`.

În cadrul funcției se vor efectua următoarele procesări. Dacă $i=j$ atunci avem $Prod[i] = outProd$. Altfel, fie $mij = (i+j)/2$. Vom calcula produsul elementelor val de pe pozițiile $i, \dots, mij$ (pornind de la $A[i]$ și $B[i]$, care sunt parametrii ai funcției). Fie acest produs PL și fie $A[mij+1]$ și $B[mij+1]$ valorile ce pot fi calculate imediat din $A[mij]$ și $B[mij]$ ($A[mij]$ și $B[mij]$ sunt obținute la sfârșitul traversării tuturor elementelor de pe pozițiile $i, \dots, mij$). Apelăm apoi `divide_et_impera(mij+1, j, A[mij+1], B[mij+1], (outProd * PL) mod Q)`. Apoi, pornind de la $A[mij+1]$ și $B[mij+1]$, vom calcula $PR = \text{produsul tuturor elementelor } val \text{ de pe pozițiile } mij+1, \dots, j \text{ (modulo } Q)$.

La final apelăm `divide_et_impera(i, mij, A[i], B[i], (outProd * PR) mod Q)`.

Complexitatea de timp a soluției este $O(N \cdot \log(N))$, iar memoria folosită este de ordinul $O(\log(N))$ (deoarece dimensiunea stivei de apeluri recursive ajunge la $\log(N)$).

Soluția 4

Complexitate de timp: $O(N)$

Memorie: $O(\sqrt{N})$

Punctaj: 100 puncte

Această soluție se bazează pe soluția 2, însă memorează vectorul PR doar din K în K poziții. Când ajungem la o poziție i , avem produsul PL al elementelor dinaintea lui i , valoarea $PR[h]$ a următoarei poziții $h > i$ memorate din vectorul PR (h este la distanță de cel mult K de i), precum și un alt vector cu valorile elementelor de pe pozițiile de la i până la h . Procesând valorile cu atenție, putem obține o complexitate liniară ($O(N)$), iar memoria folosită este de ordinul $K + N/K$. Valoarea minimă a memoriei utilizate se obține pentru $K = \sqrt{N}$.