

## Problema Resolution - descriere soluție

Fișier de intrare     `stdin`  
Fișier de ieșire     `stdout`

Autor: Iulia Groza

Problema se traduce ca:

Având un șir de  $N$  numere:  $A_1, A_2, \dots, A_N$ , se cere să se răspundă la  $M$  interogări independente  $(p, q)$ , de forma "Care este suma primelor  $q$  elemente ale șirului, după un număr nelimitat de interschimbări între elemente divizibile cu  $p$ ?".

### Subtask 1

Din moment ce toate elementele din șir sunt interschimbabile, problema se rezumă la a sorta descrescător șirul pentru a putea alege cele mai mari valori. Pentru determinarea sumelor în cazul fiecărei interogări în timp optim, se utilizează sume prefix. Se construiește în  $O(N)$  structura auxiliară  $sum[]$ , unde:

$$sum[i] = \sum_{k=1}^i A_k, i = \overline{1, N}$$

Răspunsul în  $O(1)$  pentru o interogare  $(p, q)$  este  $sum[q]$ . Complexitatea de timp totală este  $O(\max(N, M))$ .

### Subtask 2

Construim mulțimea  $D = \{A_i | A_i \text{ este par}, i = \overline{1, N}\}$  și calculăm sume prefix pentru  $D$ , utilizând structura  $presum[]$ :

$$presum[i] = \sum_{k=1}^i D_k, i = \overline{1, N}$$

Pentru a putea determina suma celor mai mari  $X$  elemente pare din șir, vom sorta crescător  $D$  și vom calcula sumele sufix corespunzătoare acesteia, notate cu  $sufsum[]$ .

Notăm cu  $X$  numărul elementelor pare din primele  $q$  elemente din  $A$ , pentru fiecare interogare. Pentru a determina numărul  $X$ , reținem indicii elementelor pare, în ordinea în care se află în  $A$ , într-o structură auxiliară  $ind[]$ . Pentru fiecare interogare  $(p, q)$ , se caută binar  $q$  în  $ind[]$ , astfel determinând numărul de elemente care trebuie interschimbate în  $A_1, A_2, \dots, A_q$ :  $X$ .

În final, răspunsul pentru fiecare interogare este  $S = sum[q] - presum[X] + sufsum[X]$ , unde  $sum[]$  este structura de sume prefix menționată în subtask-ul anterior. Complexitatea de timp totală este  $O(N + M \log N)$ .

### Subtask 3

(Autor: Răzvan Bogdan)

Pentru fiecare interogare  $(p, q)$ , se parcurge vectorul de elemente și se rețin:

- $S_{all}$  - Suma primelor  $q$  elemente;
- $S_p$  - Suma elementelor divizibile cu  $p$  din intervalul  $[1...q]$  și numărul acestora  $nr_p$ ;
- $B[]$  - Un vector cu toate elementele divizibile cu  $p$ .

Intuitiv, pentru a maximiza suma primelor  $q$  elemente, vom înlocui toate elementele divizibile cu  $p$  din intervalul  $[1...q]$  cu cele mai mari  $nr_p$  din vectorul  $B$ . Așadar, rămâne doar să se sorteze descrescător vectorul  $B$ .

În final, răspunsul este:  $S_{all} + \sum_{i=1}^{nr_p} B_i - S_p$ .

Complexitatea de timp pe interogare este  $O(N + |B| * \log|B|)$ .

Scorul maxim asociat acestui subtask poate fi obținut atât prin efectuarea unei căutări liniare, cât și prin procesarea sumelor la nivel de interogare (fără precalculare) sau descompunerea în factori primi ineficientă.

### Subtask 4

Față de cel de-al doilea subtask, nu există restricții în ceea ce privește valoarea lui  $p$ . Astfel, soluția de la subtask-ul 2 se aplică pentru toți factorii primi din care sunt compuse elementele din  $A$ . Lista de factori primi se determină în timp optim folosind Ciurul lui Eratostene.

*Observație:* Limita valorilor elementelor din  $A$  este  $Q = 10^6$ , așadar fiecare element din  $A$  poate avea maxim  $\log Q < 20$  factori primi.

Se precalculează structurile menționate în subtask-ul 2 pentru fiecare element din lista de factori primi, astfel, transformând structurile de sume parțiale în structuri bidimensionale. Precalcularea este realizată în  $O(N \log Q + Q \log \log Q)$ . Complexitatea de timp a procesării interogărilor este  $O(M \log N)$ .