Basme

Autor: stud. Ioana Gabor, Universitatea Babeș-Bolyai, Cluj-Napoca

Problema se împarte în două sub-probleme: prima dintre ele constă în determinarea sumei xor a factorilor primi pentru fiecare număr din șir, iar a doua constă în calcularea numărului de subsecvențe continue care au suma xor zero, din șirul obținut după primul pas.

Subtaskul 1: $\mathcal{O}(N \cdot sqrt(V) + N^2)$

Fiecare număr se descompune în factori primi, parcurgând de la 2 la $sqrt(n)$ posibilii divizori ai acestuia. În noul șir, fiecare element va fi suma xor a factorilor primi din vechiul șir. Pentru șirul obținut, numărul secventelor care au suma xor 0 se afla încercând toate variantele posibile. Se iterează cu i de la 1 la n, pentru capătul din stânga al secventei, iar cu j de la $i + 1$ la n, pentru capătul din dreapta al secventei. Suma xor se calculează pe parcurs.

Subtaskul 3 (toate numerele sunt prime): $\mathcal{O}(N \cdot log(N))$

Având în vedere că toate numerele sunt prime, nu mai este necesară descompunerea acestora în factori primi. Șirul "al doilea" este identic cu șirul inițial.

Operația xor are proprietatea că $a \oplus a = 0$, oricare ar fi a. Astfel, pentru a verifica dacă o subsecvența $[a, b]$ are suma xor 0, se verifica dacă suma xor a prefixului $sumPrefix[a - 1]$ este egală cu suma xor a prefixului $sumPrefix[b]$. Pentru a număra toate subsecvențele care respectă aceasta proprietate, se parcurge șirul și se calculează la fiecare pas suma prefixului curent. Într-un map se păstrează contoarele sumelor prefixelor obținute anterior. La soluție, se adaugă $countSumPrefix[currentPrefix]$, deoarece pozițiile unde suma prefixului a fost egala cu suma prefixului curent sunt capete din stânga ale secventelor de suma xor 0 care au capătul din dreapta la poziția curenta.

Subtaskul 2: $\mathcal{O}(sqrt(V) \cdot loglog(sqrt(V)) + sqrt(V)/log(sqrt(V)) \cdot log(V) \cdot N + N \cdot logN)$

Pentru prima parte, cea a descompunerii numerelor, se folosește ciurul lui Erathostene pentru generarea numerelor prime pana la $sqrt(V)$. Când dorim sa descompunem un număr x în factori primi, parcurgem numerele prime mai mici decât $sqrt(x)$ și îl împărțim la cele cu care se divide. Dacă la final rămânem cu un număr mai mare decât 1, acel număr e cu siguranță prim, deci îl adăugăm la suma xor.

Pentru a doua parte, se procedează ca la subtaskul 3.

Subtaskul 4: $\mathcal{O}(V \cdot loglogV + N \cdot logV + N \cdot logN)$

Descompunerea numerelor în factori primi se poate face mai eficient, în special pentru un număr mare de query-uri, prin precalcularea "celui mai mic factor prim"($smallestPrimeFactor$) pentru fiecare valoare de la 2 la V, cu ajutorul ciurului lui Erathostene. Pentru descompunerea unui număr x, acesta se împarte succesiv la $smallestPrimeFactor[x]$. Complexitatea unei singure descompuneri este logaritmică.

Pentru a doua parte, se procedează întocmai ca la subtaskul 3.